

# Subexponential fixed-parameter tractability of cluster editing

Fedor V. Fomin\*    Stefan Kratsch†    Marcin Pilipczuk‡    Michał Pilipczuk\*  
Yngve Villanger\*

## Abstract

In the CORRELATION CLUSTERING, also known as CLUSTER EDITING, we are given an undirected  $n$ -vertex graph  $G$  and a positive integer  $k$ . The task is to decide if  $G$  can be transformed into a cluster graph, i.e., a disjoint union of cliques, by changing at most  $k$  adjacencies, i.e. by adding/deleting at most  $k$  edges. We give a subexponential algorithm that

- in time  $2^{\mathcal{O}(\sqrt{pk})} + n^{\mathcal{O}(1)}$  decides whether  $G$  can be transformed into a cluster graph with  $p$  cliques by changing at most  $k$  adjacencies.

We complement our algorithmic findings by the following tight lower bounds on the asymptotic behaviour of our algorithm. We show that unless ETH fails

- for any constant  $0 < \sigma \leq 1$ , there is  $p = \Theta(k^\sigma)$  such that there is no algorithm deciding in time  $2^{o(\sqrt{pk})} \cdot n^{\mathcal{O}(1)}$  whether  $G$  can be transformed into a cluster graph with  $p$  cliques by changing at most  $k$  adjacencies.

## 1 Introduction

*Correlation clustering*, also known as *clustering with qualitative information*, or *cluster editing*, is the problem to cluster objects based only on qualitative information concerning similarity between pairs of items. For every pair of objects, we have an indication if the objects are similar or not. The task is to find a partition of the objects into clusters minimizing the amount of similarities between different clusters and non-similarities inside of clusters. The problem was introduced by Ben-Dor, Shamir, and Yakhini [7] motivated by some problems from computational biology, and, independently, by Bansal, Blum, and Chawla [6], motivated by machine learning problems concerning document clustering according to similarities. The correlation version of clustering was studied intensively, including [1, 4, 5, 14, 15, 24, 33].

The graph-theoretic formulation of the problem is the following. A graph  $K$  is a *cluster graph* if every connected component of  $K$  is a complete graph. Let  $G = (V, E)$  be a graph and let  $F \subset V \times V$  be such that  $G \Delta F = (V, E \Delta F)$  is a cluster graph, then  $F$  is called a *cluster editing set* for  $G$ . Here  $E \Delta F$  is the symmetric difference between  $E$  and  $F$ . In the optimization version of the problem the task is to find a cluster editing set of minimum size. Constant factor approximation algorithms for this problem were obtained in [1, 6, 14]. On the negative side, the problem is known

---

\*Department of Informatics, University of Bergen, Bergen, Norway,  
{fomin,Michal.Pilipczuk,yngve.villanger}@ii.uib.no

†Utrecht University, Utrecht, the Netherlands, s.kratsch@uu.nl

‡Institute of Informatics, University of Warsaw, Poland, malcin@mimuw.edu.pl

to be NP-complete [33]. Moreover, it was shown by Charikar, Guruswami, and Wirth [14] that the problem is APX-hard.

Giotis and Guruswami [24] initiated the study of clustering when the maximum number of clusters that we are allowed to use is stipulated to be a fixed constant  $p$ . As observed by them, this type of clustering is well-motivated in settings where the number of clusters might be an external constraint that has to be met. It appeared, that  $p$ -clustering variants posed new and non-trivial challenges. In particular, in spite of the APX-hardness of general case, Giotis and Guruswami [24] gave a PTAS for this version of the problem.

In parameterized complexity the problem was studied under the name of CLUSTER EDITING.

CLUSTER EDITING

*Input:* A graph  $G = (V, E)$  and a non-negative integer  $k$ .

*Parameter:*  $k$ .

*Question:* Is there a cluster editing set for  $G$  of size at most  $k$ ?

The parameterized version of CLUSTER EDITING, and variants of it, were studied intensively [8, 9, 10, 11, 12, 17, 20, 25, 27, 28, 31, 32]. The problem is solvable in time  $\mathcal{O}(1.62^k + n + m)$  [8] and it has a kernel with  $2k$  vertices [13, 16] (see Section 2 for the definition of a kernel).

A cluster graph  $G$  is called a  $p$ -cluster graph if it has  $p$  connected components or, equivalently, if it is a vertex-disjoint union of  $p$  cliques. Similarly, a set  $F$  is a  $p$ -cluster editing set of  $G$ , if  $G' = (V, E \triangle F)$  is a  $p$ -cluster graph. We also study the following variation of the clustering problem.

$p$ -CLUSTER EDITING

*Input:* A graph  $G = (V, E)$  and a non-negative integer  $k$ .

*Parameter:*  $k$ .

*Question:* Is there a  $p$ -cluster editing set for  $G$  of size at most  $k$ ?

Shamir et al. [33] showed that  $p$ -CLUSTER EDITING is NP-complete for every fixed  $p \geq 2$ . A kernel with  $(p + 2)k + p$  vertices was given by Guo [26].

**Subexponential complexity.** In this paper we establish several results around the (im)possibility of solving CLUSTER EDITING in subexponential time. Flum and Grohe [21, Chapter 16] defined the complexity class SUBEPT, which, loosely speaking—we skip here some technical conditions—is the class of problems solvable in time  $2^{o(k)}n^{\mathcal{O}(1)}$ , where  $n$  is the input length and  $k$  is the parameter. This is a very interesting class because the problems from SUBEPT are “easier” than “usual” parameterized problems. To make this statement more concrete, we need a well-known complexity hypothesis formulated by Impagliazzo, Paturi, and Zane [29].

**Exponential Time Hypothesis (ETH):** There is a positive real  $s$  such that 3-CNF-SAT with  $n$  variables and  $m$  clauses cannot be solved in time  $2^{sn}(n + m)^{\mathcal{O}(1)}$ .

This hypothesis is widely applied in the theory of exact exponential algorithms for hard problems, which are better than the trivial exhaustive search, though still exponential [22]. Flum and Grohe have shown that most of the natural parameterized problems are not in SUBEPT unless ETH fails [21]. Thus it is most likely that the majority of parameterized problems are not solvable

in subexponential time. Until very recently, the only problems known to be in the class SUBEPT were the problems with additional constraints on the input, like being a planar,  $H$ -minor-free, or tournament graph [3, 18]. However, recent algorithmic developments indicate that the structure of the class SUBEPT is much more interesting than expected. It appears that a class of parameterized problems related to chordal graphs, like MINIMUM FILL-IN or CHORDAL GRAPH SANDWICH, are in SUBEPT [23].

There is a striking resemblance between CLUSTER EDITING and FEEDBACK ARC SET ON TOURNAMENTS (FAST), and a number of generic algorithmic approaches can be applied for both problems [2, 4]. By a result of Alon, Lokshtanov, and Saurabh [3], FAST is in SUBEPT. Based on this, Cao and Chen [13] conjectured that CLUSTER EDITING is also solvable in subexponential time. While we resolve negatively this conjecture, a refinement of the problem, the case when the number of clusters is bounded, belongs to SUBEPT.

**Our results.** Our main result is the following theorem establishing the membership of  $p$ -CLUSTER EDITING in SUBEPT.

**Theorem 1.**  *$p$ -CLUSTER EDITING is solvable in time  $\mathcal{O}(2^{\mathcal{O}(\sqrt{pk})} + m + n)$  on instances with  $n$  vertices and  $m$  edges.*

Let us remark that by Theorem 1,  $p$ -CLUSTER EDITING is in the class SUBEPT for  $p = o(k)$ . The ideas used to prove Theorem 1 are quite different from what was used for FAST [3] or MINIMUM FILL-IN [23]. The crucial observation is that a  $p$ -cluster graph has  $2^{\mathcal{O}(\sqrt{pk})}$  cuts of size at most  $k$  (henceforth called  $k$ -cuts). As in a YES-instance to the  $p$ -CLUSTER EDITING problem, each  $k$ -cut is a  $2k$ -cut of a  $p$ -cluster graph, we infer a similar bound on the number of cuts if the input instance is a YES-instance. This allows us to use dynamic programming over the set of  $k$ -cuts of the input graph. Together with a kernelization algorithm for  $p$ -CLUSTER EDITING, this yields Theorem 1.

We complement Theorem 1 with two lower bounds. Our first lower bound is based on Theorem 2. Its subsequent Corollaries 1 and 2 show that the exponential time dependence of our algorithm is asymptotically tight for any reasonable choice of parameters  $p$  and  $k$ .

**Theorem 2.** *For any  $\varepsilon > 0$  there is  $\delta > 0$  and a polynomial-time algorithm that, given positive integers  $p$  and  $k$  and a 3-CNF-SAT formula  $\Phi$  with  $n$  variables and  $m$  clauses, such that  $k, n \geq \varepsilon p$  and  $n, m \leq \sqrt{pk}/\varepsilon$ , computes a graph  $G$  and integer  $k'$ , such that  $k' \leq \delta k$ ,  $|V(G)| \leq \delta \sqrt{pk}$  and*

- *if  $\Phi$  is satisfiable, then there exists a  $6p$ -cluster graph  $G_0$  with  $V(G) = V(G_0)$  and  $|E(G) \Delta E(G_0)| \leq k'$ ; and*
- *if there exists a  $p'$ -cluster graph  $G_0$  with  $p' \leq 6p$ ,  $V(G) = V(G_0)$  and  $|E(G) \Delta E(G_0)| \leq k'$ , then  $\Phi$  is satisfiable.*

The statement of Theorem 2 may look very technical, so let us now shortly elaborate about its consequences. Recall that an existence of a subexponential, in both the number of variables and clauses, algorithm for verifying satisfiability of 3-CNF-SAT formulas would violate ETH [29].

**Corollary 1.** *Unless ETH fails, for every  $0 < \sigma \leq 1$ , there is  $p = \Theta(k^\sigma)$  such that  $p$ -CLUSTER EDITING is not solvable in time  $2^{o(\sqrt{pk})}|V(G)|^{\mathcal{O}(1)}$ .*

*Proof.* Assume we are given a 3-CNF-SAT formula  $\Phi$  with  $n$  variables and  $m$  clauses. If  $n < m$ ,  $\lceil (m - n)/2 \rceil$  times perform the following operation: add three new variables  $x, y$  and  $z$ , and clause  $(x \vee y \vee z)$ . In this way we preserve the satisfiability of  $\Phi$ , increase the size of  $\Phi$  by a constant factor, and ensure that  $n \geq m$ .

Take now  $k = \lceil n^{\frac{2}{1+\sigma}} \rceil$ ,  $p = \lceil n^{\frac{2\sigma}{1+\sigma}} \rceil$ . As  $n \geq m$  and  $0 < \sigma \leq 1$ , we have  $k, n \geq p$  and  $n, m \leq \sqrt{pk}$  but  $n, m = \Omega(\sqrt{pk})$ . Invoke Theorem 2 for  $\varepsilon = 1$  and feed the reduction algorithm with the formula  $\Phi$  and parameters  $p$  and  $k$ , obtaining a graph  $G$  and a parameter  $k'$ . Note that  $6p = \Theta(k^\sigma)$ . Apply the assumed algorithm for the  $p$ -CLUSTER EDITING problem to the instance  $(G, 6p, k')$ . In this way we resolve the satisfiability of  $\Phi$  in time  $2^{o(\sqrt{pk})}|V(G)|^{\mathcal{O}(1)} = 2^{o(n+m)}$ , contradicting ETH.  $\square$

**Corollary 2.** *Unless ETH fails, there does not exist a constant  $p \geq 6$  and an algorithm that solves  $p$ -CLUSTER EDITING in time  $2^{o(\sqrt{k})}|V(G)|^{\mathcal{O}(1)}$  or  $2^{o(|V(G)|)}$  for a fixed number of  $p$  clusters.*

*Proof.* We prove the corollary for  $p = 6$ ; the claim for larger values of  $p$  can be easily obtained by adding  $p - 6$  large cliques to the graph obtained in the reduction.

Assume we are given a 3-CNF-SAT formula  $\Phi$  with  $n$  variables and  $m$  clauses. Take  $k = \max(n, m)^2$ , invoke Theorem 2 for  $\varepsilon = 1$  and feed the reduction algorithm with the formula  $\Phi$  and parameters 6 and  $k$ , obtaining a graph  $G$  and a parameter  $k'$ . Note that  $|V(G)| = \mathcal{O}(\sqrt{k})$ . Apply the assumed algorithm for the  $p$ -CLUSTER EDITING problem to the instance  $(G, 6, k')$ . In this way we resolve the satisfiability of  $\Phi$  in time  $2^{o(\sqrt{k})}|V(G)|^{\mathcal{O}(1)} = 2^{o(n+m)}$ , contradicting ETH.  $\square$

Let us remark that Theorem 2 and Corollary 1 do not rule out a possibility that CLUSTER EDITING is solvable in subexponential time. Our second lower bound shows that when the number of clusters is not constrained, then the problem cannot be solved in subexponential time unless ETH fails. This disproves the conjecture of Cao and Chen [13]. We note that Theorem 3 was independently obtained by Komusiewicz in his PhD thesis [30].

**Theorem 3.** *Unless ETH fails, CLUSTER EDITING cannot be solved in time  $2^{o(k)}n^{\mathcal{O}(1)}$ .*

Let us remark that the proof of Theorem 3 holds for the case when the number of clusters  $p$  is  $\Omega(k)$ . Theorems 1 and 3 show that a phase transition of the problem complexity occurs when the number of clusters switch from sublinear to linear function of  $k$ . It is also worth to note that Theorems 1 and 3 establish interesting parallels between parameterized complexity and approximability of clustering problems. As we already mentioned, bounding the number of clusters drops the complexity of the problem drastically—from APX-hardness to PTAS [14, 24]. By Theorems 1 and 3, exactly the same phenomena occurs for subexponential-time solvability of clustering.

## 2 Preliminaries

We denote by  $G = (V, E)$  a finite, undirected, and simple graph with vertex set  $V(G) = V$  and edge set  $E(G) = E$ . We also use  $n$  to denote the number of vertices and  $m$  the number of edges in  $G$ . For a nonempty subset  $W \subseteq V$ , the subgraph of  $G$  induced by  $W$  is denoted by  $G[W]$ . We say that a vertex set  $W \subseteq V$  is *connected* if  $G[W]$  is connected. The *open neighborhood* of a vertex  $v$  is  $N(v) = \{u \in V : uv \in E\}$  and the *closed neighborhood* is  $N[v] = N(v) \cup \{v\}$ . For a vertex set  $W \subseteq V$  we put  $N(W) = \bigcup_{v \in W} N(v) \setminus W$  and  $N[W] = N(W) \cup W$ . For graphs  $G, H$ , by  $\mathcal{H}(G, H)$  we denote the number of edge modifications needed to obtain  $H$  from  $G$ , i.e., the number of edges present in  $G$  and not present in  $H$  plus vice versa.

**Parameterized complexity.** A parameterized problem  $\Pi$  is a subset of  $\Gamma^* \times \mathbb{N}$  for some finite alphabet  $\Gamma$ . An instance of a parameterized problem consists of  $(x, k)$ , where  $k$  is called the parameter. A central notion in parameterized complexity is *fixed-parameter tractability (FPT)* which means, for a given instance  $(x, k)$ , solvability in time  $f(k) \cdot p(|x|)$ , where  $f$  is an arbitrary computable function of  $k$  and  $p$  is a polynomial in the input size. We refer to the book of Downey and Fellows [19] for further reading on parameterized complexity.

**Kernelization.** A *kernelization algorithm* for a parameterized problem  $\Pi \subseteq \Gamma^* \times \mathbb{N}$  is an algorithm that given  $(x, k) \in \Gamma^* \times \mathbb{N}$  outputs in time polynomial in  $|x| + k$  a pair  $(x', k') \in \Gamma^* \times \mathbb{N}$ , called a *kernel* such that  $(x, k) \in \Pi$  if and only if  $(x', k') \in \Pi$ ,  $|x'| \leq g(k)$ , and  $k' \leq k$ , where  $g$  is some computable function.

We need the following result of Guo [26].

**Proposition 4** ([26]).  *$p$ -CLUSTER EDITING admits a kernel with  $(p+2)k+p$  vertices. The running time of the kernelization algorithm is  $\mathcal{O}(n+m)$ , where  $n$  is the number of vertices and  $m$  the number of edges in the input graph  $G$*

The following lemma is used in both our lower bounds. Its proof is almost identical to the proof of Lemma 1 in [26], and we provide it here for reader's convenience.

**Lemma 5.** *Let  $G = (V, E)$  be an undirected graph and  $X \subseteq V$  be a set of vertices such that  $G[X]$  is a clique and each vertex in  $X$  has the same set of neighbors outside  $X$  (i.e.,  $N_G[v] = N_G[X]$  for each  $v \in X$ ). Let  $F \subseteq V \times V$  be a set such that  $G \triangle F$  is a cluster graph where the vertices of  $X$  are in at least two different clusters. Then there exists  $F' \subseteq V \times V$  such that: (i)  $|F'| < |F|$ , (ii)  $G \triangle F'$  is a cluster graph with no larger number of clusters than  $G \triangle F$ , (iii) in  $G \triangle F'$  the clique  $G[X]$  is contained in one cluster.*

*Proof.* For a vertex  $v \in X$ , let  $F(v) = \{u \notin X : vu \in F\}$ . Note that, since  $N_G[v] = N_G[X]$  for all  $v \in X$ , we have  $F(v) = F(w)$  if  $v$  and  $w$  belong to the same cluster in  $G \triangle F$ .

Let  $Z$  be the vertex set of a cluster in  $G \triangle F$  such that there exists  $v \in Z \cap X$  with smallest  $|F(v)|$ . Construct  $F'$  as follows: take  $F$ , and for each  $w \in X$  replace all elements of  $F$  incident with  $w$  with  $\{uw : u \in F(v)\}$ . In other words, we modify  $F$  by moving all vertices of  $X \setminus Z$  to the cluster  $Z$ . Clearly,  $G \triangle F'$  is a cluster graph,  $X$  is contained in one cluster in  $G \triangle F'$  and  $G \triangle F'$  contains no more clusters than  $G \triangle F$ . To finish the proof, we need to show that  $|F'| < |F|$ . The sets  $F$  and  $F'$  contain the same set of elements not incident with  $X$ . As  $|F(v)|$  was minimum possible, for each  $w \in X$  we have  $|F(w)| \geq |F'(w)|$ . As  $X$  was split between at least two connected components of  $G \triangle F$ ,  $F$  contains at least one edge of  $G[X]$ , whereas  $F'$  does not. We infer that  $|F'| < |F|$  and the lemma is proven.  $\square$

### 3 A subexponential algorithm for $p$ -Cluster Editing

In this section we prove Theorem 1, that is, we show a  $\mathcal{O}(2^{\mathcal{O}(\sqrt{pk})} + n + m)$ -time algorithm for  $p$ -CLUSTER EDITING.

#### 3.1 Reduction for large $p$

The initial step of our algorithm consist of simple preprocessing rules reducing the instance to an equivalent instance with  $p \leq 6k$ .

Note that at most  $2k$  vertices can be incident to the elements of  $F$ . Thus, when  $p > 2k$ , the input instance needs to contain at least  $p - 2k$  connected components that are cliques and are left untouched in  $G \triangle F$ . However, even if  $G$  contains a lot of connected components that are cliques, we may want to merge or split some of these cliques to obtain exactly  $p$  clusters.

Consider the case that  $p > 6k$ . If  $G$  contains less than  $p - 2k$  connected components that are cliques then  $(G, p, k)$  is a NO-instance. If  $G$  contains more than  $2k$  isolated vertices, at least one of these vertices is not incident to an element of  $F$ , thus we may delete one isolated vertex and decrease  $p$  by one.

**Rule 1** If  $G$  contains  $2k + 1$  isolated vertices, pick one of them, say  $v$ , and delete it from  $G$ . New instance is  $(G \setminus v, p - 1, k)$ .

We are left with the case where  $G$  contains more than  $2k$  connected components that are cliques, but not isolated vertices. At least one of these cliques does not contain a vertex that is incident to an element of  $F$ . As discussed above, the cliques may be merged with other vertices (to decrease the number of connected components), or split into more clusters (to increase the number of connected components). In both cases, we can greedily merge or split the smallest possible cluster. Thus, without loss of generality, we can assume that the largest connected component of  $G$  that is a clique is left untouched in  $G \triangle F$ . We reduce the input instance  $(G, p, k)$  by deleting this cluster and decreasing  $p$  by one.

**Rule 2** If  $G$  contains  $2k + 1$  isolated cliques that are not isolated vertices, pick a clique  $C$  of largest size and delete it from  $G$ . The new instance is  $(G \setminus C, p - 1, k)$ .

By the arguments above, every YES-instance  $(G, p, k)$  for which none of the reduction rules applies, has  $p \leq 6k$ . For the rest of this section we assume that  $p \leq 6k$ .

### 3.2 Binomial coefficient bounds

Before we proceed with the description of the core part of the algorithm, we need several purely mathematical technical bounds.

**Lemma 6.** *If  $a, b$  are positive integers, then  $\binom{a+b}{a} \leq \frac{(a+b)^{a+b}}{a^a b^b}$ .*

*Proof.* In the proof we use a folklore fact that the sequence  $a_n = (1 + 1/n)^n$  is increasing. This implies that  $(1 + \frac{1}{b})^b \leq (1 + \frac{1}{a+b})^{a+b}$ , equivalently  $\frac{(a+b)^{a+b}}{b^b} \leq \frac{(a+b+1)^{a+b}}{(b+1)^b}$ .

Let us fix  $a$ ; we prove the claim via induction with respect to  $b$ . For  $b = 1$  the claim is equivalent to  $a^a \leq (a+1)^a$  and therefore trivial. In order to check the induction step, notice that

$$\begin{aligned} \binom{a+b+1}{a} &= \frac{a+b+1}{b+1} \cdot \binom{a+b}{a} \leq \frac{a+b+1}{b+1} \cdot \frac{(a+b)^{a+b}}{a^a b^b} \\ &\leq \frac{a+b+1}{b+1} \cdot \frac{(a+b+1)^{a+b}}{a^a (b+1)^b} = \frac{(a+b+1)^{a+b+1}}{a^a (b+1)^{b+1}}. \end{aligned}$$

□

**Lemma 7.** *If  $a, b$  are nonnegative integers, then  $\binom{a+b}{a} \leq 2^{2\sqrt{ab}}$ .*

*Proof.* Firstly, observe that the claim is trivial for  $a = 0$  or  $b = 0$ ; hence, we can assume that  $a, b > 0$ . Moreover, without losing generality assume that  $a \leq b$ . Let us denote  $\sqrt{ab} = \ell$  and  $\frac{a}{b} = t$ , then  $0 < t \leq 1$ . By Lemma 6 we have that

$$\begin{aligned} \binom{a+b}{a} &\leq \frac{(a+b)^{a+b}}{a^a b^b} = \left(1 + \frac{b}{a}\right)^a \cdot \left(1 + \frac{a}{b}\right)^b \\ &= \left[ \left(1 + \frac{1}{t}\right)^{\frac{\sqrt{t}}{1}} \cdot \left(1 + \frac{t}{1}\right)^{\frac{1}{\sqrt{t}}} \right]^\ell. \end{aligned}$$

Let us denote  $g(x) = x \ln(1 + x^{-2}) + x^{-1} \ln(1 + x^2)$ . As  $\binom{a+b}{a} \leq e^{\ell \cdot g(\sqrt{t})}$ , it suffices to prove that  $g(x) \leq 2 \ln 2$  for all  $0 < x \leq 1$ . Observe that

$$\begin{aligned} g'(x) &= \ln(1 + x^{-2}) - x \cdot 2x^{-3} \cdot \frac{1}{1 + x^{-2}} - x^{-2} \ln(1 + x^2) + x^{-1} \cdot 2x \cdot \frac{1}{1 + x^2} \\ &= \ln(1 + x^{-2}) - \frac{2}{1 + x^2} - x^{-2} \ln(1 + x^2) + \frac{2}{1 + x^2} \\ &= \ln(1 + x^{-2}) - x^{-2} \ln(1 + x^2). \end{aligned}$$

Let us now introduce  $h : (0, 1] \rightarrow \mathbb{R}$ ,  $h(y) = g'(\sqrt{y}) = \ln(1 + y^{-1}) - y^{-1} \ln(1 + y)$ . Then,

$$\begin{aligned} h'(y) &= -y^{-2} \cdot \frac{1}{1 + y^{-1}} + y^{-2} \ln(1 + y) - y^{-1} \cdot \frac{1}{1 + y} \\ &= y^{-2} \ln(1 + y) - \frac{2}{y + y^2}. \end{aligned}$$

We claim that  $h'(y) \leq 0$  for all  $y \leq 1$ . Indeed, from the inequality  $\ln(1 + y) \leq y$  we infer that

$$y^{-2} \ln(1 + y) \leq y^{-1} = \frac{2}{y + y} \leq \frac{2}{y + y^2}.$$

Therefore,  $h'(y) \leq 0$  for  $y \in (0, 1]$ , so  $h(y)$  is non-increasing on this interval. As  $h(1) = 0$ , this implies that  $h(y) \geq 0$  for  $y \in (0, 1]$ , so also  $g'(x) \geq 0$  for  $x \in (0, 1]$ . Theat means that  $g(x)$  is non-decreasing on the interval  $(0, 1]$ , so  $g(x) \leq g(1) = 2 \ln 2$ .  $\square$

### 3.3 Small cuts

We can now slowly proceed to the algorithm itself. Let us introduce the key notion.

**Definition 8.** Let  $G = (V, E)$  be an undirected graph. A partition  $(V_1, V_2)$  of  $V$  is called a  $k$ -nice partition of  $G$  if  $|E(V_1, V_2)| \leq k$ .

Firstly, we observe that  $k$ -nice partitions can be quickly enumerated.

**Lemma 9.**  $k$ -nice partitions of a graph  $G$  can be enumerated with polynomial delay.

*Proof.* We follow the standard branching. We order the vertices arbitrarily, start with empty  $V_1$ ,  $V_2$  and for each consecutive vertex  $v$  we branch into two subcases: we put  $v$  either into  $V_1$  or into  $V_2$ . Once the alignment of all vertices is decided, we output the partition. However, each time we

put a vertex in one of the sides, we run a polynomial-time max-flow algorithm to check whether the minimum edge cut between  $V_1$  and  $V_2$  constructed so far is at most  $k$ . If not, then we terminate this branch as it certainly results in no solutions found. Thus, we always pursue a branch that results in at least one feasible solution, and finding the next solution occurs within a polynomial number of steps.  $\square$

Intuitively,  $k$ -nice partitions of the graph  $G$  form the search space of the algorithm. Therefore, we would like to bound their number. This we do by firstly bounding the number of nice partitions of a cluster graph, and then using the fact that a YES-instance is not very far from some cluster graph.

**Lemma 10.** *Let  $K$  be a cluster graph containing at most  $p$  clusters, where  $p \leq 6k$ . Then the number of  $k$ -nice partitions of  $K$  is at most  $2^{8\sqrt{pk}}$ .*

*Proof.* By somewhat abusing the notation, assume that  $K$  has exactly  $p$  clusters, some of which may be empty. Let  $C_1, C_2, \dots, C_p$  be these clusters and  $c_1, c_2, \dots, c_p$  be their sizes, respectively. We firstly establish a bound on the number of partitions  $(V_1, V_2)$  such that the cluster  $C_i$  contains  $x_i$  vertices from  $V_1$  and  $y_i$  from  $V_2$ . Then we discuss that the number of ways of selecting pairs  $x_i, y_i$  summing up to  $c_i$ , for which the number of  $k$ -nice partitions is positive, can be bounded in a similar manner. Multiplying the obtained two bounds gives us the claim.

Having fixed the numbers  $x_i, y_i$ , the number of ways in which the cluster  $C_i$  can be partitioned is equal to  $\binom{x_i + y_i}{x_i}$ . Note that  $\binom{x_i + y_i}{x_i} \leq 2^{2\sqrt{x_i y_i}}$  by Lemma 7. Observe that there are  $x_i y_i$  edges between  $V_1$  and  $V_2$  inside the cluster  $C_i$ , so if  $(V_1, V_2)$  is a  $k$ -nice partition, then  $\sum_{i=1}^p x_i y_i \leq k$ . By applying the Cauchy-Schwarz inequality we infer that  $\sum_{i=1}^p \sqrt{x_i y_i} \leq \sqrt{p} \cdot \sqrt{\sum_{i=1}^p x_i y_i} \leq \sqrt{pk}$ . Therefore, the number of considered nice partitions is bounded by

$$\prod_{i=1}^p \binom{x_i + y_i}{x_i} \leq 2^{2 \sum_{i=1}^p \sqrt{x_i y_i}} \leq 2^{2\sqrt{pk}}.$$

Moreover, observe that  $\min(x_i, y_i) \leq \sqrt{x_i y_i}$ ; hence,  $\sum_{i=1}^p \min(x_i, y_i) \leq \sqrt{pk}$ . Therefore, the choice of  $x_i, y_i$  can be modeled by first choosing for each  $i$ , whether  $\min(x_i, y_i)$  is equal to  $x_i$  or to  $y_i$ , and then expressing  $\lfloor \sqrt{pk} \rfloor$  as the sum of  $p + 1$  nonnegative numbers:  $\min(x_i, y_i)$  for  $1 \leq i \leq p$  and the rest,  $\lfloor \sqrt{pk} \rfloor - \sum_{i=1}^p \min(x_i, y_i)$ . The number of choices in the first step is equal to  $2^p \leq 2^{\sqrt{6pk}}$ , and in the second is equal to  $\binom{\lfloor \sqrt{pk} \rfloor + p}{p} \leq 2^{\sqrt{pk} + \sqrt{6pk}}$ . Therefore, the number of possible choices of  $x_i, y_i$  is bounded by  $2^{(1+2\sqrt{6})\sqrt{pk}} \leq 2^{6\sqrt{pk}}$ . Hence, the total number of  $k$ -nice partitions is bounded by  $2^{6\sqrt{pk}} \cdot 2^{2\sqrt{pk}} = 2^{8\sqrt{pk}}$ , as claimed.  $\square$

**Lemma 11.** *If  $(G, p, k)$  is a YES-instance of  $p$ -CLUSTER EDITING with  $p \leq 6k$ , then the number of  $k$ -nice partitions of  $G$  is bounded by  $2^{8\sqrt{2pk}}$ .*

*Proof.* Let  $K$  be a cluster graph with at most  $p$  clusters such that  $\mathcal{H}(G, K) \leq k$ . Observe that every  $k$ -nice partition of  $G$  is also a  $2k$ -nice partition of  $K$ , as  $K$  differs from  $G$  by at most  $k$  edge modifications. The claim follows from Lemma 10.  $\square$

### 3.4 The algorithm

We are now ready to prove Theorem 1.



*Proof of Theorem 1.* Let  $(G = (V, E), p, k)$  be the given  $p$ -CLUSTER EDITING instance, and let  $(V, \overline{E})$  be the complement of the graph  $G$ . By making use of Proposition 4, we assume that  $G$  has at most  $(p+2)k+p$  vertices, thus all the factors polynomial in the size of  $G$  can be henceforth hidden within the  $2^{\mathcal{O}(\sqrt{pk})}$  factor. Application of Proposition 4 gives the additional  $\mathcal{O}(n+m)$  summand to the complexity. By further using the reduction rules introduced in Section 3.1 we can also assume that  $p \leq 6k$ .

We now enumerate the  $k$ -nice partitions of  $G$  with polynomial delay. If we exceed the bound  $2^{8\sqrt{2pk}}$  given by Lemma 11, we know that we can safely answer NO, so we immediately terminate the computation and give a negative answer. Therefore, we can assume that we have computed the set  $\mathcal{N}$  of all  $k$ -nice partitions of  $G$  and  $|\mathcal{N}| \leq 2^{8\sqrt{2pk}}$ .

Assume that  $(G, p, k)$  is a YES-instance and let  $K$  be a cluster graph with at most  $p$  clusters such that  $\mathcal{H}(G, K) \leq k$ . Again, let  $C_1, C_2, \dots, C_p$  be the clusters of  $K$ , where, by somewhat abusing the notation, some of these clusters can be empty. Observe that for every  $j \in \{0, 1, 2, \dots, p\}$ , the partition  $(\bigcup_{i=1}^j V(C_i), \bigcup_{i=j+1}^p V(C_i))$  has to be  $k$ -nice, as otherwise there would be more than  $k$  edges that need to be deleted from  $G$  in order to obtain  $K$ . This observation enables us to use a dynamic programming approach on the set of nice partitions.

We construct a directed graph  $D$ , whose vertex set is equal to  $\mathcal{N} \times \{0, 1, 2, \dots, p\} \times \{0, 1, 2, \dots, k\}$ ; note that  $|V(D)| = 2^{\mathcal{O}(\sqrt{pk})}$ . We create arcs going from  $((V_1, V_2), j, \ell)$  to  $((V'_1, V'_2), j+1, \ell')$ , where  $V_1 \subseteq V'_1$  (hence  $V_2 \supseteq V'_2$ ),  $j \in \{0, 1, 2, \dots, p-1\}$  and  $\ell' = \ell + |E(V_1, V'_1 \setminus V_1)| + |\overline{E}(V'_1 \setminus V_1, V_1 \setminus V_1)|$ . The arcs can be constructed in  $2^{\mathcal{O}(\sqrt{pk})}$  time by checking for all the pairs of vertices whether they should be connected. We claim that the answer to the instance  $(G, p, k)$  is equivalent to reachability of any of the vertices of form  $((V, \emptyset), p, \ell)$  from the vertex  $((\emptyset, V), 0, 0)$ .

In one direction, if there is a path from  $((\emptyset, V), 0, 0)$  to  $((V, \emptyset), p, \ell)$  for some  $\ell \leq k$ , then the consecutive sets  $V'_1 \setminus V_1$  along the path form clusters  $C_i$  of a cluster graph  $K$ , whose editing distance to  $G$  is accumulated on the last coordinate, thus bounded by  $k$ . In the second direction, if there is a cluster graph  $K$  with clusters  $C_1, C_2, \dots, C_p$  within editing distance at most  $k$  from  $G$ , then vertices  $((\bigcup_{i=1}^j V(C_i), \bigcup_{i=j+1}^p V(C_i)), j, \mathcal{H}(G[\bigcup_{i=1}^j V(C_i)], K[\bigcup_{i=1}^j V(C_i)]))$  form a path from  $((\emptyset, V), 0, 0)$  to  $((V, \emptyset), p, \mathcal{H}(G, K))$ . Note that all these triples are indeed vertices of the graph  $D$ , as  $(\bigcup_{i=1}^j V(C_i), \bigcup_{i=j+1}^p V(C_i))$  are  $k$ -nice partitions of  $G$ .

Reachability in a directed graph can be tested in linear time with respect to the number of vertices and arcs, using, for example, breadth-first search. We can now apply this algorithm to the graph  $D$  and conclude solving the  $p$ -CLUSTER EDITING instance in  $\mathcal{O}(2^{\mathcal{O}(\sqrt{pk})} + n + m)$  time.  $\square$

## 4 Multivariate lower bound

This section is devoted to the proof of Theorem 2. The proof consists of four parts. In Section 4.1 we preprocess the input formula  $\Phi$  to make it more regular. Section 4.2 contains the details of the construction of the graph  $G$ . In Section 4.3 we show how to translate a satisfying assignment of  $\Phi$  into a  $6p$ -cluster graph  $G_0$  close to  $G$  and we provide a reverse implication in Section 4.4. In the proof we treat  $\varepsilon$  as a constant and hide the factors depending on it in the  $\mathcal{O}$ -notation. That is, the constants in the  $\mathcal{O}$ -notation correspond to the factor  $\delta$  in the statement of Theorem 2.

## 4.1 Preprocessing of the formula

We start with a step that regularizes the input formula  $\Phi$ , while increasing its size only by a constant factor. The purpose of this step is to ensure that, when we translate a satisfying assignment of  $\Phi$  into a cluster graph  $G_0$  in the completeness step, the clusters are of the same size, and therefore contain the minimum possible number of edges. This property is used in the argumentation of the soundness step.

**Lemma 12.** *For any fixed  $\varepsilon > 0$ , there exists a polynomial-time algorithm that, given a 3-CNF formula  $\Phi$  with  $n$  variables and  $m$  clauses and an integer  $p$ ,  $\varepsilon p \leq n$ , constructs a 3-CNF formula  $\Phi'$  with  $n'$  variables and  $m'$  clauses together with a partition of the variable set  $\text{Vars}(\Phi')$  into  $p$  parts  $\text{Vars}^r$ ,  $1 \leq r \leq p$ , such that following properties hold:*

- (a)  $\Phi'$  is satisfiable iff  $\Phi$  is;
- (b) in  $\Phi'$  every clause contains exactly three literals corresponding to different variables;
- (c) in  $\Phi'$  every variable appears exactly three times positively and exactly three times negatively;
- (d)  $n'$  is divisible by  $p$  and, for each  $1 \leq r \leq p$ ,  $|\text{Vars}^r| = n'/p$  (i.e., the variables are split evenly between the parts  $\text{Vars}^r$ );
- (e) if  $\Phi'$  is satisfiable, then there exists a satisfying assignment of  $\text{Vars}(\Phi')$  with the property that in each part  $\text{Vars}^r$  the same number of variables is set to true as to false.
- (f)  $n' + m' = \mathcal{O}(n + m)$ , where the constant hidden in the  $\mathcal{O}$ -notation depends on  $\varepsilon$ .

*Proof.* We modify  $\Phi$  while preserving satisfiability, consecutively ensuring that properties (b), (c), (d), and (e) are satisfied. Satisfaction of (f) will follow directly from the constructions used.

First, delete every clause that contains two different literals corresponding to the same variable, as they are always satisfied. Remove copies of the same literals inside clauses. Until all the clauses have at least two literals, remove every clause containing one literal, set the value of this literal so that the clause is satisfied and propagate this knowledge to the other clauses. At the end, create a new variable  $p$  and for every clause  $C$  that has two literals replace it with two clauses  $C \vee p$  and  $C \vee \neg p$ . All these operations preserve satisfiability and at the end all the clauses consist of exactly three different literals.

Second, duplicate each clause so that every variable appears an even number of times. Introduce two new variables  $q, r$ . Take any variable  $x$ , assume that  $x$  appears positively  $k^+$  times and negatively  $k^-$  times. If  $k^+ < k^-$ , introduce clauses  $(x \vee q \vee r)$  and  $(x \vee \neg q \vee \neg r)$ , each  $\frac{k^- - k^+}{2}$  times, otherwise introduce clauses  $(\neg x \vee q \vee r)$  and  $(\neg x \vee \neg q \vee \neg r)$ , each  $\frac{k^+ - k^-}{2}$  times. These operations preserve satisfiability (as the new clauses can be satisfied by setting  $q$  to true and  $r$  to false) and, after the operation, every variable appears the same number of times positively as negatively (including the new variables  $q, r$ ).

Third, copy each clause three times. For each variable  $x$ , replace all occurrences of the variable  $x$  with a cycle of implications in the following way. Assume that  $x$  appears  $6d$  times (the number of appearances is divisible by six due to the modifications in the previous paragraph and the copying step). Introduce new variables  $x_i$  for  $1 \leq i \leq 3d$ ,  $y_i$  for  $1 \leq i \leq d$  and clauses  $(\neg x_i \vee x_{i+1} \vee y_{\lceil i/3 \rceil})$  and  $(\neg x_i \vee x_{i+1} \vee \neg y_{\lceil i/3 \rceil})$  for  $1 \leq i \leq 3d$  (with  $x_{3d+1} = x_1$ ). Moreover, replace each occurrence of the variable  $x$  with one of the variables  $x_i$  in such a way that each variable  $x_i$  is used once in a

positive literal and once in a negative one. In this manner each variable  $x_i$  and  $y_i$  is used exactly three times in a positive literal and three times in a negative one. Moreover, the new clauses form an implication cycle  $x_1 \Rightarrow x_2 \Rightarrow \dots \Rightarrow x_{3d} \Rightarrow x_1$ , ensuring that all the variables  $x_i$  have equal value in any satisfying assignment of the formula.

Fourth, to make  $n'$  divisible by  $p$  we first copy the entire formula three times, creating a new set of variables for each copy. In this way we ensure that the number of variables is divisible by three. Then we add new variables in triples to make the number of variables divisible by  $p$ . For each triple  $x, y, z$  of new variables, we introduce six new clauses: all possible clauses that contain one literal for each variable  $x, y$  and  $z$  except for  $(x \vee y \vee z)$  and  $(\neg x \vee \neg y \vee \neg z)$ . Note that the new clauses are easily satisfied by setting all new variables to true, while all new variables appear exactly three times positively and three times negatively. Moreover, as initially  $\varepsilon p \leq n$ , this step increases the size of the formula only by a constant factor.

Finally, to achieve (d) and (e) take  $\Phi' = \Phi \wedge \bar{\Phi}$ , where  $\bar{\Phi}$  is a copy of  $\Phi$  on a disjoint copy of the variable set and with all literals reversed, i.e., positive occurrences are replaced by negative ones and vice versa. Of course, if  $\Phi'$  is satisfiable then  $\Phi$  as well, while if  $\Phi$  is satisfiable, then we can copy the assignment to the copies of variables and reverse it, thus obtaining a feasible assignment for  $\Phi'$ . Recall that before this step the number of variables was divisible by  $p$ . We can now partition the variable set into  $p$  parts, such that whenever we include a variable into one part, we include its copy in the same part as well. In order to prove that the property (e) holds, take any feasible solution to  $\Phi'$ , truncate the evaluation to  $\text{Vars}(\Phi)$  and copy it while reversing on  $\bar{\Phi}$ .  $\square$

## 4.2 Construction

In this section we show how to compute the graph  $G$  and the integer  $k'$  from the formula  $\Phi'$  given by Lemma 12. As Lemma 12 increases the size of the formula by a constant factor, we have that  $n', m' = \mathcal{O}(\sqrt{pk})$  and  $|\text{Vars}^r| = n'/p = \mathcal{O}(\sqrt{k/p})$  for  $1 \leq r \leq p$ .

Observe that in the statement of the Theorem 2 we can safely assume that  $\varepsilon \leq 1$ , as the assumptions become more and more restricted as  $\varepsilon$  becomes smaller. From now on we assume that  $\varepsilon \leq 1$ .

Let  $L = 1000 \cdot \left(1 + \frac{n'}{p\varepsilon}\right) = \mathcal{O}(\sqrt{k/p})$ . For each part  $\text{Vars}^r$ ,  $1 \leq r \leq p$ , we create six cliques  $Q_\alpha^r$ ,  $1 \leq \alpha \leq 6$ , each of size  $L$ . Let  $\mathcal{Q}$  be the set of all vertices of all cliques  $Q_\alpha^r$ . In this manner we have  $6p$  cliques. Intuitively, if we seek for a  $6p$ -cluster graph close to  $G$ , then the cliques are large enough so that merging two cliques is expensive — in the intended solution we have exactly one clique in each cluster.

For every variable  $x \in \text{Vars}^r$  we create six vertices  $w_{1,2}^x, w_{2,3}^x, \dots, w_{5,6}^x, w_{6,1}^x$ . Connect them into a cycle in this order; this cycle is called a *6-cycle for the variable  $x$* . Moreover, for each  $1 \leq \alpha \leq 6$  and  $v \in V(Q_\alpha^r)$ , create edges  $vw_{\alpha-1,\alpha}^x$  and  $vw_{\alpha,\alpha+1}^x$  (we assume that the indices behave cyclicly, i.e.,  $w_{6,7}^x = w_{6,1}^x$ ,  $Q_7^r = Q_1^r$  etc.). Let  $\mathcal{W}$  be the set of all vertices  $w_{\alpha,\alpha+1}^x$  for all variables  $x$ . Intuitively, the cheapest way to cut the 6-cycle for variable  $x$  is to assign the vertices  $w_{\alpha,\alpha+1}^x$ ,  $1 \leq \alpha \leq 6$  all either to the clusters with cliques with only odd indices or only with even indices. Choosing even indices corresponds to setting  $x$  to false, while choosing odd ones corresponds to setting  $x$  to true.

Let  $r(x)$  be the index of the part that contains the variable  $x$ , that is,  $x \in \text{Vars}^{r(x)}$ .

In each clause  $C$  we (arbitrarily) enumerate variables: for  $1 \leq \eta \leq 3$ , let  $\text{var}(C, \eta)$  be the variable in the  $\eta$ -th literal of  $C$ , and  $\text{sgn}(C, \eta) = 0$  if the  $\eta$ -th literal is negative and  $\text{sgn}(C, \eta) = 1$  otherwise.

For every clause  $C$  create nine vertices:  $s_{\beta,\xi}^C$  for  $1 \leq \beta, \xi \leq 3$ . The edges incident to the vertex  $s_{\beta,\xi}^C$  are defined as follows:

- for each  $1 \leq \eta \leq 3$  create an edge  $s_{\beta,\xi}^C w_{2\beta+2\eta-3, 2\beta+2\eta-2}^{\text{var}(C,\eta)}$ ;
- if  $\xi = 1$ , for each  $1 \leq \eta \leq 3$  connect  $s_{\beta,\xi}^C$  to all vertices of one of the cliques the vertex  $w_{2\beta+2\eta-3, 2\beta+2\eta-2}^{\text{var}(C,\eta)}$  is adjacent to depending on the sign of the  $\eta$ -th literal in  $C$ , that is, the clique  $Q_{2\beta+2\eta-2-\text{sgn}(C,\eta)}^{r(\text{var}(C,\eta))}$ ;
- if  $\xi > 1$ , for each  $1 \leq \eta \leq 3$  connect  $s_{\beta,\xi}^C$  to all vertices of both cliques the vertex  $w_{2\beta+2\eta-3, 2\beta+2\eta-2}^{\text{var}(C,\eta)}$  is adjacent to, that is, the cliques  $Q_{2\beta+2\eta-3}^{r(\text{var}(C,\eta))}$  and  $Q_{2\beta+2\eta-2}^{r(\text{var}(C,\eta))}$ .

We note that for a fixed vertex  $s_{\beta,\xi}^C$ , the aforementioned cliques  $s_{\beta,\xi}^C$  is adjacent to are pairwise different, and they have pairwise different subscripts (but may have equal superscripts, i.e., belong to the same part). See Figure 1 for an illustration.

Let  $\mathcal{S}$  be the set of all vertices  $s_{\beta,\xi}^C$  for all clauses  $C$ . If we seek a  $6p$ -cluster graph close to the graph  $G$ , it is reasonable to put a vertex  $s_{\beta,\xi}^C$  in a cluster together with one of the cliques this vertex is attached to. If  $s_{\beta,\xi}^C$  is put in a cluster together with one of the vertices  $w_{2\beta+2\eta-3, 2\beta+2\eta-2}^{\text{var}(C,\eta)}$  for  $1 \leq \eta \leq 3$ , we do not need to cut the appropriate edge. The vertices  $s_{\beta,1}^C$  verify the assignment encoded by the variable vertices  $w_{\alpha,\alpha+1}^x$ ; the vertices  $s_{\beta,2}^C$  and  $s_{\beta,3}^C$  help us to make all clusters be of equal size (which is helpful in the soundness argument).

We note that  $|V(G)| = 6pL + \mathcal{O}(n' + m') = \mathcal{O}(\sqrt{pk})$ .

We now define the budget  $k'$  for edge editings. To make the presentation more clear, we split this budget into few summands. Let

$$\begin{aligned} k_{\mathcal{Q}-\mathcal{Q}} &= 0, & k_{\mathcal{Q}-\mathcal{WS}} &= (6n' + 36m')L, & k_{\mathcal{WS}-\mathcal{WS}}^{\text{all}} &= 6p \binom{6n'+9m'}{6p} \frac{1}{2}, \\ k_{\mathcal{WS}-\mathcal{WS}}^{\text{exist}} &= 6n' + 27m', & k_{\mathcal{W}-\mathcal{W}}^{\text{save}} &= 3n', & k_{\mathcal{W}-\mathcal{S}}^{\text{save}} &= 9m' \end{aligned}$$

and finally

$$k' = k_{\mathcal{Q}-\mathcal{Q}} + k_{\mathcal{Q}-\mathcal{WS}} + k_{\mathcal{WS}-\mathcal{WS}}^{\text{all}} + k_{\mathcal{WS}-\mathcal{WS}}^{\text{exist}} - 2k_{\mathcal{W}-\mathcal{W}}^{\text{save}} - 2k_{\mathcal{W}-\mathcal{S}}^{\text{save}}.$$

Note that, as  $p \leq k$ ,  $L = \mathcal{O}(\sqrt{k/p})$  and  $n', m' = \mathcal{O}(\sqrt{pk})$ , we have  $k' = \mathcal{O}(k)$ .

The intuition behind this split is as follows. The intended solution for the  $p$ -CLUSTER EDITING instance  $(G, 6p, k')$  creates no edges between the cliques  $Q_\alpha^r$ , each clique is contained in its own cluster, and  $k_{\mathcal{Q}-\mathcal{Q}} = 0$ . For each  $v \in \mathcal{W} \cup \mathcal{S}$ , the vertex  $v$  is assigned to a cluster with one clique  $v$  is adjacent to;  $k_{\mathcal{Q}-\mathcal{WS}}$  accumulates the cost of removal of other edges in  $E(\mathcal{Q}, \mathcal{W} \cup \mathcal{S})$ . Finally, we count the editings in  $(\mathcal{W} \cup \mathcal{S}) \times (\mathcal{W} \cup \mathcal{S})$  in an indirect way. First we cut all edges of  $E(\mathcal{W} \cup \mathcal{S}, \mathcal{W} \cup \mathcal{S})$  (summand  $k_{\mathcal{WS}-\mathcal{WS}}^{\text{exist}}$ ). We group the vertices of  $\mathcal{W} \cup \mathcal{S}$  into clusters and add edges between vertices in each cluster; the summand  $k_{\mathcal{WS}-\mathcal{WS}}^{\text{all}}$  corresponds to the cost of this operation when all the clusters are of the same size (and the number of edges is minimum possible). Finally, in summands  $k_{\mathcal{W}-\mathcal{W}}^{\text{save}}$  and  $k_{\mathcal{W}-\mathcal{S}}^{\text{save}}$  we count how many edges are removed and then added again in this process:  $k_{\mathcal{W}-\mathcal{W}}^{\text{save}}$  corresponds to saving three edges from each 6-cycle in  $E(\mathcal{W}, \mathcal{W})$  and  $k_{\mathcal{W}-\mathcal{S}}^{\text{save}}$  corresponds to saving one edge in  $E(\mathcal{W}, \mathcal{S})$  per each vertex  $s_{\beta,\xi}^C$ .

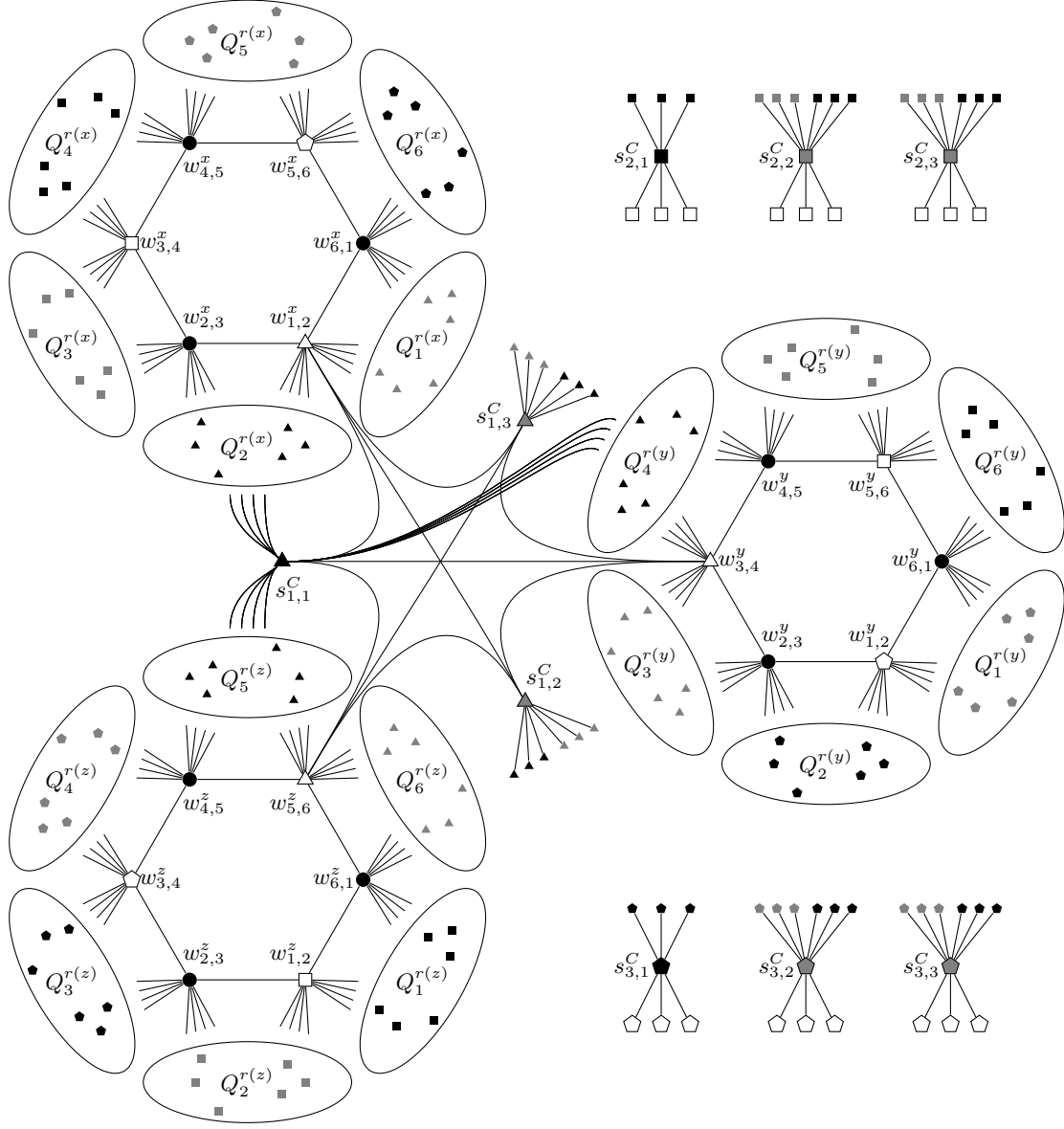


Figure 1: A part of the graph  $G$  created for the clause  $C = (\neg x \vee \neg y \vee z)$ , with  $\text{var}(C, 1) = x$ ,  $\text{var}(C, 2) = y$  and  $\text{var}(C, 3) = z$ . Note that the parts  $r(x)$ ,  $r(y)$  and  $r(z)$  may be not pairwise distinct. However, due to the rotation index  $\beta$ , in any case for a fixed vertex  $s_{\beta, \xi}^C$  the cliques this vertex is adjacent to on this figure are pairwise distinct and have pairwise distinct subscripts.

### 4.3 Completeness

We now show how to translate a satisfying assignment of the input formula  $\Phi$  into a  $6p$ -cluster graph close to  $G$ .

**Lemma 13.** *If the input formula  $\Phi$  is satisfiable, then there exists a  $6p$ -cluster graph  $G_0$  on vertex set  $V(G)$  such that  $\mathcal{H}(G, G_0) = k'$ .*

*Proof.* Let  $\phi'$  be a satisfying assignment of the formula  $\Phi'$  as guaranteed by Lemma 12. Recall that in each part  $\text{Vars}^r$ , the assignment  $\phi'$  sets the same number of variables to true as to false.

To simplify the presentation, we identify the range of  $\phi'$  with integers:  $\phi'(x) = 0$  if  $x$  is evaluated to false in  $\phi'$  and  $\phi'(x) = 1$  otherwise. Moreover, for a clause  $C$  by  $\eta(C)$  we denote the index of an arbitrarily chosen literal that satisfies  $C$  in the assignment  $\phi'$ .

We create  $6p$  clusters  $K_\alpha^r$ ,  $1 \leq r \leq p$ ,  $1 \leq \alpha \leq 6$ , as follows:

- $Q_\alpha^r \subseteq K_\alpha^r$  for  $1 \leq r \leq p$ ,  $1 \leq \alpha \leq 6$ ;
- for  $x \in \text{Vars}(\Phi')$ , if  $\phi'(x) = 1$  then  $w_{6,1}^x, w_{1,2}^x \in K_1^{r(x)}$ ,  $w_{2,3}^x, w_{3,4}^x \in K_3^{r(x)}$ ,  $w_{4,5}^x, w_{5,6}^x \in K_5^{r(x)}$ ;
- for  $x \in \text{Vars}(\Phi')$ , if  $\phi'(x) = 0$  then  $w_{1,2}^x, w_{2,3}^x \in K_2^{r(x)}$ ,  $w_{3,4}^x, w_{4,5}^x \in K_4^{r(x)}$ ,  $w_{5,6}^x, w_{6,1}^x \in K_6^{r(x)}$ ;
- for each clause  $C$  of  $\Phi'$  and  $1 \leq \beta, \xi \leq 3$  we define  $\eta = \eta(C) + \xi - 1$  and we assign the vertex  $s_{\beta,\xi}^C$  to the cluster  $K_{2\beta+2\eta-2-\phi'(\text{var}(C,\eta))}^{r(\text{var}(C,\eta))}$ .

Note that in this way  $s_{\beta,\xi}^C$  belongs to the same cluster as its neighbor  $w_{2\beta+2\eta-3, 2\beta+2\eta-2}^{\text{var}(C,\eta)}$ . See Figure 2 for an illustration.

Let us now compute  $\mathcal{H}(G, G_0)$ . We do not need to add nor delete any edges in  $G[\mathcal{Q}]$ . We note that each vertex  $v \in \mathcal{W} \cup \mathcal{S}$  is assigned to a cluster with one clique  $Q_\alpha^r$  it is adjacent to. Indeed, this is only non-trivial for vertices  $s_{\beta,1}^C$  for clauses  $C$  and  $1 \leq \beta \leq 3$ . Note that this vertex belongs to the same cluster as the vertex  $w_{2\beta+2\eta(C)-2-\phi'(\text{var}(C,\eta(C)))}^{\text{var}(C,\eta(C))}$ , and, since the  $\eta(C)$ -th literal of  $C$  satisfies  $C$  in the assignment  $\phi'$ ,  $s_{\beta,1}^C$  is adjacent to all vertices of the clique  $Q_{2\beta+2\eta(C)-2-\phi'(\text{var}(C,\eta(C)))}^{r(\text{var}(C,\eta(C)))}$ .

Therefore we need to cut  $k_{\mathcal{Q}-\mathcal{WS}} = (6n' + 36m')L$  edges in  $E(\mathcal{Q}, \mathcal{W} \cup \mathcal{S})$ :  $L$  edges adjacent to each vertex  $w_{\alpha,\alpha+1}^x$ ,  $2L$  edges adjacent to each vertex  $s_{\beta,1}^C$ , and  $5L$  edges adjacent to each vertex  $s_{\beta,2}^C$  and  $s_{\beta,3}^C$ . We do not add any new edges between  $\mathcal{Q}$  and  $\mathcal{W} \cup \mathcal{S}$ .

To count the number of editings in  $G[\mathcal{W} \cup \mathcal{S}]$ , let us first verify that the clusters  $K_\alpha^r$  are of equal sizes. Fix cluster  $K_\alpha^r$ ,  $1 \leq \alpha \leq 6$ ,  $1 \leq r \leq p$ .  $K_\alpha^r$  contains two vertices  $w_{\alpha-1,\alpha}^x$  and  $w_{\alpha,\alpha+1}^x$  for each variable  $x$  with  $\phi'(x) + \alpha$  being even. Since  $\phi'$  evaluates the same number of variables in  $\text{Vars}^r$  to true as to false, we infer that each cluster  $K_\alpha^r$  contains exactly  $n'/p$  vertices from  $\mathcal{W}$ , corresponding to  $n'/(2p) = |\text{Vars}^r|/2$  variables of  $\text{Vars}^r$ .

For  $1 \leq \alpha \leq 6$ , let  $\text{Vars}_\alpha^r = \phi^{-1}(0) \cap \text{Vars}^r$  if  $\alpha$  is even and  $\phi^{-1}(1) \cap \text{Vars}^r$  if  $\alpha$  is odd. That is,  $x \in \text{Vars}_\alpha^r$  if and only if  $w_{\alpha-1,\alpha}^x, w_{\alpha,\alpha+1}^x \in K_\alpha^r$ . By the properties of  $\Phi'$ , for each  $x \in \text{Vars}_\alpha^r$  the variable  $x$  appears in three clauses positively and in three clauses negatively; in particular, it satisfies exactly three clauses in the assignment  $\phi'$ . We claim that  $K_\alpha^r \cap \mathcal{S}$  consists of  $3|\text{Vars}_\alpha^r| = \frac{3}{2}|\text{Vars}^r|$  vertices, that is, for each variable  $x \in \text{Vars}_\alpha^r$ , for each clause  $C$  (out of three) that  $x$  satisfies in the assignment  $\phi'$ ,  $K_\alpha^r$  contains exactly one (out of nine) vertex  $s_{\beta,\xi}^C$ , and no more vertices of  $\mathcal{S}$ .

In one direction, take a variable  $x \in \text{Vars}_\alpha^r$  and a clause  $C$  that is satisfied by  $x$  in the assignment  $\phi'$ . Let  $\alpha' = 2\lceil \alpha/2 \rceil$ , so that  $w_{\alpha'-1,\alpha'}^x \in \{w_{\alpha-1,\alpha}^x, w_{\alpha,\alpha+1}^x\}$  is the vertex with first subscript odd and

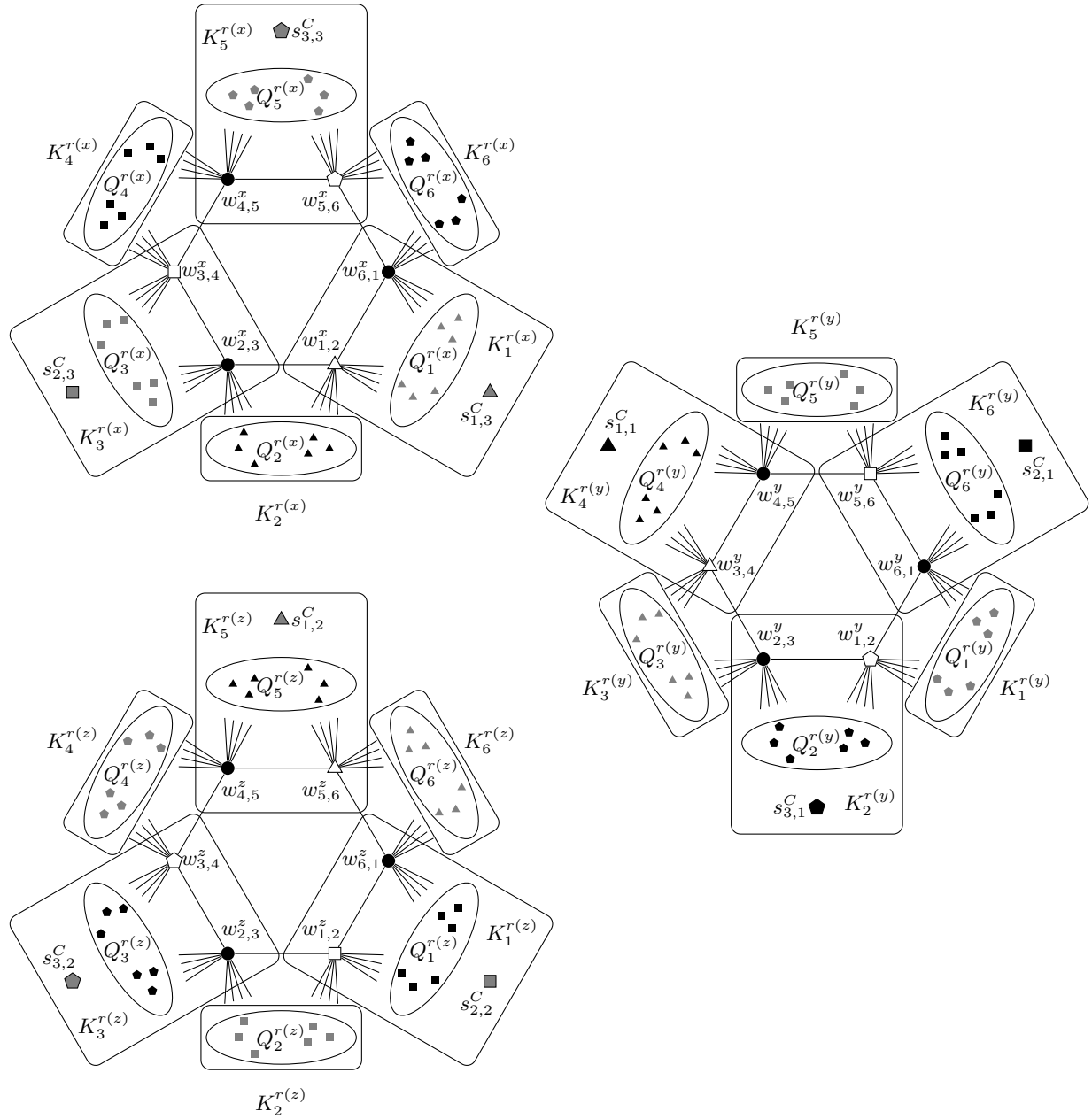


Figure 2: Parts of clusters for variables  $x$ ,  $y$  and  $z$  with  $\phi'(x) = 1$ ,  $\phi'(y) = 0$ ,  $\phi'(z) = 1$ , and a clause  $C = (\neg x \vee \neg y \vee z)$  with  $\text{var}(C, 1) = x$ ,  $\text{var}(C, 2) = y$ ,  $\text{var}(C, 3) = z$  and  $\eta(C) = 2$  (note that both  $y$  and  $z$  satisfy  $C$  in the assignment  $\phi'$ , but  $y$  was chosen as a representative).

the second even. Take  $\eta$  such that  $x = \text{var}(C, \eta)$  and  $\beta = \alpha'/2 - \eta + 1$ . Then  $\alpha' = 2\beta + 2\eta - 2$ , and the three vertices  $s_{\beta, \xi}^C$  for  $1 \leq \xi \leq 3$  are adjacent to  $w_{\alpha'-1, \alpha'}^x$ . Now let  $\xi = \eta - \eta(C) + 1$ ; then  $s_{\beta, \xi}^C$  is assigned to the same cluster as  $w_{\alpha'-1, \alpha'}^x$  since  $\eta = \eta(C) + \xi - 1$ . Since  $x \in \text{Vars}_\alpha^r$ , then  $s_{\beta, \xi}^C \in K_\alpha^r$ .

In the other direction, let  $s_{\beta, \xi}^C \in K_\alpha^r$  for some clause  $C$  and  $1 \leq \beta, \xi \leq 3$ . Recall that  $s_{\beta, \xi}^C$  belongs to the same cluster as one of its three neighbors in  $\mathcal{W}$ . Therefore there exists  $w_{\alpha'-1, \alpha'}^x$  adjacent to  $s_{\beta, \xi}^C$  that belongs to  $K_\alpha^r$ ; note that  $\alpha'$  is even. Moreover, as  $s_{\beta, \xi}^C$  and  $w_{\alpha'-1, \alpha'}^x$  are assigned to the same cluster, we infer that  $x$  satisfies  $C$ . As  $w_{\alpha'-1, \alpha'}^x \in K_\alpha^r$ , then  $x \in \text{Vars}_\alpha^r$ . Let  $\eta$  be such that  $x = \text{var}(C, \eta)$ . As  $s_{\beta, \xi}^C w_{\alpha'-1, \alpha'}^x \in E(G)$ , we have  $\alpha'/2 = \beta + \eta - 1$ , that is,  $\beta = \alpha'/2 - \eta + 1$ . Recall that the neighbors of  $s_{\beta, \xi}^C$  from  $\mathcal{W}$  have pairwise different subscripts; that is,  $s_{\beta, \xi}^C$  is adjacent to  $w_{\alpha'-1, \alpha'}^x$ ,  $w_{\alpha'+1, \alpha'+2}^{\text{var}(C, \eta+1)}$  and  $w_{\alpha'+3, \alpha'+4}^{\text{var}(C, \eta+2)}$ . Therefore the cliques that are adjacent to  $w_{\alpha'+1, \alpha'+2}^{\text{var}(C, \eta+1)}$  and  $w_{\alpha'+3, \alpha'+4}^{\text{var}(C, \eta+2)}$  are different from  $K_\alpha^r$ , and these vertices do not belong to  $K_\alpha^r$ . We infer that if  $s_{\beta, \xi}^C \in K_\alpha^r$ , that is,  $s_{\beta, \xi}^C$  belongs to the same cluster as  $w_{\alpha'-1, \alpha'}^x$ , then  $\eta = \eta(C) + \xi - 1$ ; equivalently,  $\xi = \eta - \eta(C) + 1$ . Hence,  $s_{\beta, \xi}^C \in K_\alpha^r$  only if  $C$  is satisfied by a variable in  $\text{Vars}_\alpha^r$  and, providing this, for at most one choice of the indices  $1 \leq \beta, \xi \leq 3$ . This concludes the proof of the claim.

We now count the number of editings in  $G[\mathcal{W} \cup \mathcal{S}]$  as sketched in the construction section. The subgraph  $G[\mathcal{W} \cup \mathcal{S}]$  contains  $6n' + 27m'$  edges: one 6-cycle for each variable and three edges incident to each of the nine vertices  $s_{\beta, \xi}^C$  for each clause  $C$ . Each cluster  $K_\alpha^r$  contains  $n'/p$  vertices from  $\mathcal{W}$  and  $\frac{3m'}{2p}$  vertices from  $\mathcal{S}$ . If we deleted all edges in  $G[\mathcal{W} \cup \mathcal{S}]$  and then added all the missing edges in the clusters, we would make  $k_{\mathcal{WS}-\mathcal{WS}}^{\text{exist}} + k_{\mathcal{WS}-\mathcal{WS}}^{\text{all}}$  editings, due to the clusters being equal-sized. However, in this manner we sometimes delete an edge and then introduce it again; thus, for each edge of  $G[\mathcal{W} \cup \mathcal{S}]$  that is contained in one cluster  $K_\alpha^r$ , we should subtract 2 in this counting scheme.

For each variable  $x$ , exactly three edges of the form  $w_{\alpha-1, \alpha}^x w_{\alpha, \alpha+1}^x$  are contained in one cluster; this gives a total of  $k_{\mathcal{W}-\mathcal{W}}^{\text{save}} = 3n'$  saved edges. For each clause  $C$  each vertex  $s_{\beta, \xi}^C$  is assigned to a cluster with one of the vertices  $w_{2\beta+2\eta-3, 2\beta+2\eta-2}^{\text{var}(C, \eta)}$ ,  $1 \leq \eta \leq 3$ , thus exactly one of the edges incident to  $s_{\beta, \xi}^C$  is contained in one cluster. This sums up to  $k_{\mathcal{W}-\mathcal{S}}^{\text{save}} = 9m'$  saved edges, and we infer that the  $6p$ -cluster graph  $G_0$  can be obtained from  $G$  by exactly  $k' = k_{\mathcal{Q}-\mathcal{Q}} + k_{\mathcal{Q}-\mathcal{WS}} + k_{\mathcal{WS}-\mathcal{WS}}^{\text{exist}} + k_{\mathcal{WS}-\mathcal{WS}}^{\text{all}} - 2k_{\mathcal{W}-\mathcal{W}}^{\text{save}} - 2k_{\mathcal{W}-\mathcal{S}}^{\text{save}}$  editings.  $\square$

#### 4.4 Soundness

We need the following simple bound on the number of edges of a cluster graph.

**Lemma 14.** *Let  $a, b$  be positive integers and  $H$  be a cluster graph with  $ab$  vertices and at most  $a$  clusters. Then  $|E(H)| \geq a \binom{b}{2}$  and equality holds if and only if  $H$  is an  $a$ -cluster graph and each cluster of  $H$  has size exactly  $b$ .*

*Proof.* It suffices to note that if not all clusters of  $H$  are of size  $b$ , there is one of size at least  $b + 1$  and one of size at most  $b - 1$  or the number of clusters is less than  $a$ ; then, moving a vertex from the largest cluster of  $H$  to a new or the smallest cluster strictly decreases the number of edges of  $H$ .  $\square$

We are now ready to show how to translate a  $p'$ -cluster graph  $G_0$  with  $p' \leq 6p$ ,  $\mathcal{H}(G_0, G) \leq k'$  into a satisfying assignment of the input formula  $\Phi$ .

**Lemma 15.** *If there exists a  $p'$ -cluster graph  $G_0$  with  $V(G) = V(G_0)$ ,  $p' \leq 6p$ ,  $\mathcal{H}(G, G_0) \leq k'$ , then the formula  $\Phi$  is satisfiable.*



*Proof.* By Lemma 5, we may assume that each clique  $Q_\alpha^r$  is contained in one cluster in  $G_0$ . Let  $F = E(G_0) \triangle E(G)$  be the editing set,  $|F| \leq k'$ .

Before we start, we present some intuition. The cluster graph  $G_0$  may differ from the one constructed in the completeness step in two significant ways, both leading to some savings in the edges of  $G[\mathcal{W} \cup \mathcal{S}]$  that may not be included in  $F$ . First, it may not be true that each cluster contains exactly one clique  $Q_\alpha^r$ . However, since the number of cliques is at most  $6p$ , this may happen only if some clusters contain more than one clique  $Q_\alpha^r$ , and we need to add  $L^2$  edges to merge each pair of cliques that belong to the same cluster. Second, a vertex  $v \in \mathcal{W} \cup \mathcal{S}$  may not be contained in a cluster together with one of the cliques it is adjacent to. However, as each such vertex needs to be separated from *all* its adjacent clusters (compared to *all but one* in the completeness step), this costs us additional  $L$  edges to remove. The large constant in front of the definition of  $L$  ensures us that in both these ways we pay more than we save on the edges of  $G[\mathcal{W} \cup \mathcal{S}]$ . We now proceed to the formal argumentation.

We define the following quantities.

$$\begin{aligned} \ell_{\mathcal{Q}-\mathcal{Q}} &= |F \cap (\mathcal{Q} \times \mathcal{Q})|, & \ell_{\mathcal{Q}-\mathcal{WS}} &= |F \cap E_G(\mathcal{Q}, \mathcal{W} \cup \mathcal{S})|, \\ \ell_{\mathcal{WS}-\mathcal{WS}}^{\text{all}} &= |E(G_0(\mathcal{W} \cup \mathcal{S}))|, \\ \ell_{\mathcal{W}-\mathcal{W}}^{\text{save}} &= |E_G(\mathcal{W}, \mathcal{W}) \cap E_{G_0}(\mathcal{W}, \mathcal{W})| & \ell_{\mathcal{W}-\mathcal{S}}^{\text{save}} &= |E_G(\mathcal{W}, \mathcal{S}) \cap E_{G_0}(\mathcal{W}, \mathcal{S})|. \end{aligned}$$

Recall that  $k_{\mathcal{WS}-\mathcal{WS}}^{\text{exist}} = |E(G(\mathcal{W} \cup \mathcal{S}))| = 6n' + 27m'$ . Similarly as in the completeness proof, we have that

$$|F| \geq \ell_{\mathcal{Q}-\mathcal{Q}} + \ell_{\mathcal{Q}-\mathcal{WS}} + \ell_{\mathcal{WS}-\mathcal{WS}}^{\text{all}} + k_{\mathcal{WS}-\mathcal{WS}}^{\text{exist}} - 2\ell_{\mathcal{W}-\mathcal{W}}^{\text{save}} - 2\ell_{\mathcal{W}-\mathcal{S}}^{\text{save}}.$$

Indeed,  $\ell_{\mathcal{Q}-\mathcal{Q}}$  and  $\ell_{\mathcal{Q}-\mathcal{WS}}$  count (possibly not all) edges of  $F$  that are incident to the vertices of  $\mathcal{Q}$ . The edges of  $F \cap ((\mathcal{W} \cup \mathcal{S}) \times (\mathcal{W} \cup \mathcal{S}))$  are counted in an indirect way: each edge of  $G[\mathcal{W} \cup \mathcal{S}]$  is deleted ( $k_{\mathcal{WS}-\mathcal{WS}}^{\text{exist}}$ ) and each edge of  $G_0[\mathcal{W} \cup \mathcal{S}]$  is added ( $\ell_{\mathcal{WS}-\mathcal{WS}}^{\text{all}}$ ). Then, the edges that are counted twice in this manner are subtracted ( $\ell_{\mathcal{W}-\mathcal{W}}^{\text{save}}$  and  $\ell_{\mathcal{W}-\mathcal{S}}^{\text{save}}$ ).

We say that a cluster is *crowded* if it contains at least two cliques  $Q_\alpha^r$  and *proper* if it contains exactly one clique  $Q_\alpha^r$ . A clique  $Q_\alpha^r$  that is contained in a crowded (proper) cluster is called a *crowded (proper) clique*.

Let  $a$  be the number of crowded cliques. Note that

$$\ell_{\mathcal{Q}-\mathcal{Q}} - k_{\mathcal{Q}-\mathcal{Q}} = |F \cap (\mathcal{Q} \times \mathcal{Q})| - 0 \geq aL^2/2,$$

as each vertex in a crowded clique needs to be connected to at least one other crowded clique.

We say that a vertex  $v \in \mathcal{W} \cup \mathcal{S}$  is *attached* to a clique  $Q_\alpha^r$ , if it is adjacent to all vertices of the clique in  $G$ . Moreover, we say that a vertex  $v \in \mathcal{W} \cup \mathcal{S}$  is *alone* if it is contained in a cluster in  $G_0$  that does not contain any clique  $v$  is attached to. Let  $n^{\text{alone}}$  be the number of alone vertices.

Let us now count the number of vertices a fixed clique  $Q_\alpha^r$  is attached to. Recall that  $|\text{Vars}^r| = n'/p$ . For each variable  $x \in \text{Vars}^r$  the clique  $Q_\alpha^r$  is attached to two vertices  $w_{\alpha-1, \alpha}^x$  and  $w_{\alpha, \alpha+1}^x$ . Moreover, each variable  $x \in \text{Vars}^r$  appears in exactly six clauses: thrice positively and thrice negatively. For each such clause  $C$ ,  $Q_\alpha^r$  is attached to the vertex  $s_{\beta, 2}^C$  for exactly one choice of the value  $1 \leq \beta \leq 3$  and to the vertex  $s_{\beta, 3}^C$  for exactly one choice of the value  $1 \leq \beta \leq 3$ . Moreover, if  $x$  appears in  $C$  positively and  $\alpha$  is odd, or if  $x$  appears in  $C$  negatively and  $\alpha$  is even, then  $Q_\alpha^r$  is attached to the vertex  $s_{\beta, 1}^C$  for exactly one choice of the value  $1 \leq \beta \leq 3$ . We infer that the clique

$Q_\alpha^r$  is attached to exactly fifteen vertices from  $\mathcal{S}$  for each variable  $x \in \text{Vars}^r$ . Therefore, there are exactly  $17|\text{Vars}^r| = 17n'/p$  vertices of  $\mathcal{W} \cup \mathcal{S}$  attached to  $Q_\alpha^r$ :  $2n'/p$  from  $\mathcal{W}$  and  $15n'/p$  from  $\mathcal{S}$ .

Take an arbitrary vertex  $v \in \mathcal{W} \cup \mathcal{S}$  and assume that  $v$  is attached to  $b_v$  cliques, and  $a_v$  out of them are crowded. As  $F$  needs to contain all edges of  $G$  that connect  $v$  with cliques that belong to a different cluster than  $v$ , we infer that  $|F \cap E_G(\{v\}, \mathcal{Q})| \geq (b_v - \max(1, a_v))L$ . Moreover, if  $v$  is alone,  $|F \cap E_G(\{v\}, \mathcal{Q})| \geq b_v L \geq 1 \cdot L + (b_v - \max(1, a_v))L$ . Hence

$$\begin{aligned} \ell_{\mathcal{Q}-\mathcal{WS}} = |F \cap E_G(\mathcal{Q}, \mathcal{W} \cup \mathcal{S})| &\geq n^{\text{alone}}L + \sum_{v \in \mathcal{W} \cup \mathcal{S}} (b_v - \max(1, a_v))L \\ &\geq n^{\text{alone}}L + \sum_{v \in \mathcal{W} \cup \mathcal{S}} (b_v - 1)L - \sum_{v \in \mathcal{W} \cup \mathcal{S}} a_v L. \end{aligned}$$

Recall that  $\sum_{v \in \mathcal{W} \cup \mathcal{S}} (b_v - 1)L = k_{\mathcal{Q}-\mathcal{WS}}$ . Therefore, using the fact that each clique is attached to exactly  $17n'/p$  vertices of  $\mathcal{W} \cup \mathcal{S}$ , we obtain that

$$\ell_{\mathcal{Q}-\mathcal{WS}} - k_{\mathcal{Q}-\mathcal{WS}} = |F \cap E_G(\mathcal{Q}, \mathcal{W} \cup \mathcal{S})| - k_{\mathcal{Q}-\mathcal{WS}} \geq n^{\text{alone}}L - \sum_{v \in \mathcal{W} \cup \mathcal{S}} a_v L \geq n^{\text{alone}}L - 17aLn'/p.$$

In  $G_0$ , the vertices of  $\mathcal{W} \cup \mathcal{S}$  are split between  $p' \leq 6p$  clusters and there are  $6n' + 9m'$  of them. By Lemma 14, the minimum number of edges of  $G_0[\mathcal{W} \cup \mathcal{S}]$  is attained when all clusters are of equal size and the number of clusters is maximum possible. We infer that  $\ell_{\mathcal{WS}-\mathcal{WS}}^{\text{all}} \geq k_{\mathcal{WS}-\mathcal{WS}}^{\text{all}}$ .

We are left with  $\ell_{\mathcal{W}-\mathcal{W}}^{\text{save}}$  and  $\ell_{\mathcal{W}-\mathcal{S}}^{\text{save}}$ . Recall that  $k_{\mathcal{W}-\mathcal{W}}^{\text{save}}$  counts three edges out of each 6-cycle constructed per variable of  $\Phi'$ ,  $|k_{\mathcal{W}-\mathcal{W}}^{\text{save}}| = 3n'$ , whereas  $k_{\mathcal{W}-\mathcal{S}}^{\text{save}}$  counts one edge per each vertex  $s_{\beta, \xi}^C \in \mathcal{S}$ ,  $k_{\mathcal{W}-\mathcal{S}}^{\text{save}} = 9m' = |\mathcal{S}|$ .

Consider a crowded cluster  $K$  with  $c > 1$  crowded cliques. We say that  $K$  *interferes* with a vertex  $v \in \mathcal{W} \cup \mathcal{S}$  if  $v$  is attached to a clique in  $K$ . As each clique is attached to exactly  $17n'/p$  vertices of  $\mathcal{W} \cup \mathcal{S}$ ,  $2n'/p$  belonging to  $\mathcal{W}$  and  $15n'/p$  to  $\mathcal{S}$ , in total at most  $2an'/p$  vertices of  $\mathcal{W}$  interfere with a crowded cluster and at most  $15an'/p$  vertices of  $\mathcal{S}$ .

Fix a variable  $x \in \text{Vars}(\Phi')$ . If none of the vertices  $w_{\alpha, \alpha+1}^x \in \mathcal{W}$  interferes with any crowded cluster  $K$ , then all the cliques  $Q_{\alpha'}^{r(x)}$ ,  $1 \leq \alpha' \leq 6$ , are proper cliques, each contained in a different cluster in  $G_0$ . Moreover, if additionally no vertex  $w_{\alpha, \alpha+1}^x$ ,  $1 \leq \alpha \leq 6$ , is alone, then in the 6-cycle constructed for the variable  $x$  at most three edges are not in  $F$ . On the other hand, if some of the vertices  $w_{\alpha, \alpha+1}^x \in \mathcal{W}$  interfere with a crowded cluster  $K$ , or at least one of them is alone, it may happen that all six edges of this 6-cycle are contained in one cluster of  $G_0$ . The total number of 6-cycles that contain either alone vertices or vertices interfering with crowded clusters is bounded by  $n^{\text{alone}} + an'/p$ , as every clique is attached to exactly  $n'/p$  6-cycles. In  $k_{\mathcal{W}-\mathcal{W}}^{\text{save}}$  we counted three edges per a 6-cycle, while in  $\ell_{\mathcal{W}-\mathcal{W}}^{\text{save}}$  we counted at most three edges per every 6-cycles except 6-cycles that either contain alone vertices or vertices attached to crowded cliques, for which we counted at most six edges. Hence, we infer that

$$\ell_{\mathcal{W}-\mathcal{W}}^{\text{save}} - k_{\mathcal{W}-\mathcal{W}}^{\text{save}} \leq 3(n^{\text{alone}} + an'/p).$$

We claim that if a vertex  $s_{\beta, \xi}^C \in \mathcal{S}$  (i) is not alone, and (ii) is not attached to a crowded clique, and (iii) is not adjacent to any alone vertex in  $\mathcal{W}$ , then at most one edge from  $E(\{s_{\beta, \xi}^C\}, \mathcal{W})$  may not be in  $F$ . Recall that  $s_{\beta, \xi}^C$  has exactly three neighbors in  $\mathcal{W}$ , each of them attached to exactly two cliques and all these six cliques are pairwise distinct; moreover,  $s_{\beta, \xi}^C$  is attached only to these

six cliques, if  $\beta = 2, 3$ , or only to three out of these six, if  $\beta = 1$ . Observe that (i) and (ii) imply that  $s_{\beta,\xi}^C$  is in the same cluster as exactly one of the six cliques attached to his neighbors in  $\mathcal{W}$ , so if it was in the same cluster as two of his neighbors in  $\mathcal{W}$ , then at least one of them would be alone, contradicting (iii). However, if at least one of (i), (ii) or (iii) is not satisfied, then all three edges incident to  $s_{\beta,\xi}^S$  may be contained in one cluster. As each vertex in  $\mathcal{W}$  is adjacent to at most 18 vertices in  $\mathcal{S}$  (at most 3 per every clause in which the variable is present), there are at most  $18n^{\text{alone}}$  vertices  $s_{\beta,\xi}^C$  that are alone or adjacent to an alone vertex in  $\mathcal{W}$ . Note also that the number of vertices of  $\mathcal{S}$  interfering with crowded clusters is bounded by  $15an'/p$ , as each of  $a$  crowded cliques has exactly  $15n'/p$  vertices of  $\mathcal{S}$  attached. Thus, we are able to bound the number of vertices of  $\mathcal{S}$  for which (i), (ii) or (iii) does not hold. As in  $k_{\mathcal{W}-\mathcal{S}}^{\text{save}}$  we counted one edge per every vertex of  $\mathcal{S}$ , while in  $\ell_{\mathcal{W}-\mathcal{S}}^{\text{save}}$  we counted at most one edge per every vertex of  $\mathcal{S}$  except vertices not satisfying (i), (ii), or (iii), for which we counted at most three edges, we infer that

$$\ell_{\mathcal{W}-\mathcal{S}}^{\text{save}} - k_{\mathcal{W}-\mathcal{S}}^{\text{save}} \leq 2(18n^{\text{alone}} + 15an'/p).$$

Summing up all the bounds:

$$\begin{aligned} |F| - k' &\geq (\ell_{\mathcal{Q}-\mathcal{Q}} - k_{\mathcal{Q}-\mathcal{Q}}) + (\ell_{\mathcal{Q}-\mathcal{W}\mathcal{S}} - k_{\mathcal{Q}-\mathcal{W}\mathcal{S}}) + (\ell_{\mathcal{W}\mathcal{S}-\mathcal{W}\mathcal{S}}^{\text{all}} - k_{\mathcal{W}\mathcal{S}-\mathcal{W}\mathcal{S}}^{\text{all}}) \\ &\quad - 2(\ell_{\mathcal{W}-\mathcal{W}}^{\text{save}} - k_{\mathcal{W}-\mathcal{W}}^{\text{save}}) - 2(\ell_{\mathcal{W}-\mathcal{S}}^{\text{save}} - k_{\mathcal{W}-\mathcal{S}}^{\text{save}}) \\ &\geq aL^2/2 + n^{\text{alone}}L - 17aLn'/p + 0 - 6(n^{\text{alone}} + an'/p) - 4(15n^{\text{alone}} + 18an'/p) \\ &\geq a + n^{\text{alone}} \geq 0 \end{aligned}$$

The second to last inequality follows from the choice of the value of  $L$ ,  $L = 1000 \cdot \left(1 + \frac{n'}{p\varepsilon}\right)$ ; note that in particular  $L \geq 1000$ .

We infer that  $a = 0$ , that is, each clique  $Q_\alpha^r$  is contained in a different cluster of  $G_0$ , and each cluster of  $G_0$  contains exactly one such clique. Moreover,  $n^{\text{alone}} = 0$ , that is, each vertex  $v \in \mathcal{W} \cup \mathcal{S}$  is contained in a cluster with at least one clique  $v$  is attached to; as all cliques are proper,  $v$  is contained in a cluster with exactly one clique  $v$  is attached to and  $\ell_{\mathcal{Q}-\mathcal{W}\mathcal{S}} = k_{\mathcal{Q}-\mathcal{W}\mathcal{S}}$ .

Recall that  $|F \cap ((\mathcal{W} \cup \mathcal{S}) \times (\mathcal{W} \cup \mathcal{S}))| = \ell_{\mathcal{W}\mathcal{S}-\mathcal{W}\mathcal{S}}^{\text{all}} + k_{\mathcal{W}\mathcal{S}-\mathcal{W}\mathcal{S}}^{\text{exist}} - 2\ell_{\mathcal{W}-\mathcal{W}}^{\text{save}} - 2\ell_{\mathcal{W}-\mathcal{S}}^{\text{save}}$ . As each clique is now proper and no vertex is alone, for each variable  $x$  at most three edges out of the 6-cycle  $w_{\alpha,\alpha+1}^x$ ,  $1 \leq \alpha \leq 6$ , are not in  $F$ , that is,  $\ell_{\mathcal{W}-\mathcal{W}}^{\text{save}} \leq k_{\mathcal{W}-\mathcal{W}}^{\text{save}}$ . Moreover, for each vertex  $s_{\beta,\xi}^C \in \mathcal{S}$ , the three neighbors of  $s_{\beta,\xi}^C$  are contained in different clusters and at most one edge incident to  $s_{\beta,\xi}^C$  is not in  $F$ , that is,  $\ell_{\mathcal{W}-\mathcal{S}}^{\text{save}} \leq k_{\mathcal{W}-\mathcal{S}}^{\text{save}}$ . As  $|F| \leq k'$ , these inequalities are tight: exactly three edges out of each 6-cycle are not in  $F$ , and exactly one edge adjacent to a vertex in  $\mathcal{S}$  is not in  $F$ .

Consider an assignment  $\phi'$  of  $\text{Vars}(\Phi')$  that assigns  $\phi'(x) = 1$  if the vertices  $w_{\alpha,\alpha+1}^x$ ,  $1 \leq \alpha \leq 6$  are contained in clusters with cliques  $Q_1^{r(x)}$ ,  $Q_3^{r(x)}$ , and  $Q_5^{r(x)}$  (i.e., the edges  $w_{6,1}^x w_{1,2}^x$ ,  $w_{2,3}^x w_{3,4}^x$  and  $w_{4,5}^x w_{5,6}^x$  are not in  $F$ ), and  $\phi'(x) = 0$  otherwise (i.e., if the vertices  $w_{\alpha,\alpha+1}^x$ ,  $1 \leq \alpha \leq 6$  are contained in clusters with cliques  $Q_2^{r(x)}$ ,  $Q_4^{r(x)}$  and  $Q_6^{r(x)}$ ) — a direct check shows that these are the only ways to save 3 edges inside a 6-cycle. We claim that  $\phi'$  satisfies  $\Phi'$ . Consider a clause  $C$ . The vertex  $s_{1,1}^C$  is contained in a cluster with one of the three cliques it is attached to (as  $n^{\text{alone}} = 0$ ), say  $Q_{\alpha'}^r$ , and with one of the three vertices of  $\mathcal{W}$  it is adjacent to, say  $w_{\alpha,\alpha+1}^x$ . Therefore  $r(x) = r$ ,  $w_{\alpha,\alpha+1}^x$  is contained in the same cluster as  $Q_{\alpha'}^r$ , and  $\phi'(x)$  satisfies the clause  $C$ .  $\square$

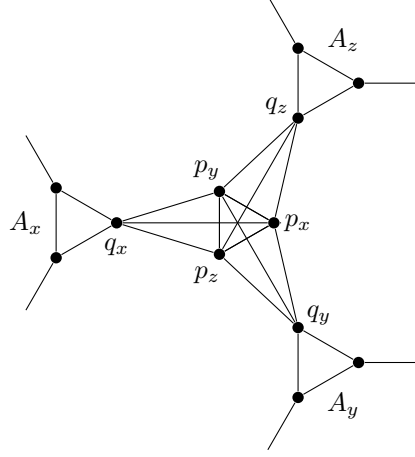


Figure 3: The gadget for a clause  $C$  with variables  $x, y$  and  $z$ .

## 5 General clustering under ETH

In this section we prove Theorem 3, namely that the CLUSTER EDITING problem without restriction on the number of clusters in the output does not admit a  $2^{o(k)} n^{\mathcal{O}(1)}$  algorithm unless the Exponential Time Hypothesis fails.

The following lemma provides a linear reduction from the problem of verifying satisfiability of 3-CNF formulas.

**Lemma 16.** *There exists a polynomial-time algorithm that, given a 3-CNF formula  $\Phi$  with  $n$  variables and  $m$  clauses, constructs a CLUSTER EDITING instance  $(G, k)$  such that (i)  $\Phi$  is satisfiable if and only if  $(G, k)$  is a YES-instance, and (ii)  $|V(G)| + |E(G)| + k = \mathcal{O}(n + m)$ .*

*Proof.* By standard arguments, we may assume that each clause of  $\Phi$  consists of exactly three literals with different variables and each variable appears at least twice: at least once in a positive literal and at least once in a negative one. Let  $\text{Vars}(\Phi)$  denote the set of variables of  $\Phi$ . For a variable  $x$ , let  $s_x$  be the number of appearances of  $x$  in the formula  $\Phi$ . For a clause  $C$  with variables  $x, y$ , and  $z$ , we denote by  $l_{x,C}$  the literal of  $C$  that contains  $x$  (i.e.,  $l_{x,C} = x$  or  $l_{x,C} = \neg x$ ).

**Construction.** We construct a graph  $G = (V, E)$  as follows. First, for each variable  $x$  we introduce a cycle  $A_x$  of length  $4s_x$ . For each clause  $C$  where  $x$  appears we assign four consecutive vertices  $a_{x,C}^j$ ,  $1 \leq j \leq 4$  on the cycle  $A_x$ . If the vertices assigned to a clause  $C'$  follow the vertices assigned to a clause  $C$  on the cycle  $A_x$ , we let  $a_{x,C'}^5 = a_{x,C}^1$ .

Second, for each clause  $C$  with variables  $x, y$ , and  $z$  we introduce a gadget with 6 vertices  $V_C = \{p_x, p_y, p_z, q_x, q_y, q_z\}$  with all inner edges except for  $q_x q_y$ ,  $q_y q_z$ , and  $q_z q_x$  (see Figure 3). If  $l_{x,C} = x$  then we connect  $q_x$  to the vertices  $a_{x,C}^1$  and  $a_{x,C}^2$ , and if  $l_{x,C} = \neg x$ , we connect  $q_x$  to  $a_{x,C}^2$  and  $a_{x,C}^3$ . We proceed analogously for variables  $y$  and  $z$  in the clause  $C$ . We set  $k = 8m + 2 \sum_{x \in \text{Vars}(\Phi)} s_x = 14m$ . This finishes the construction of the CLUSTER EDITING instance  $(G, k)$ . Clearly  $|V(G)| + |E(G)| + k = \mathcal{O}(n + m)$ . We now prove that  $(G, k)$  is a YES-instance if and only if  $\Phi$  is satisfiable.

**Completeness.** Assume that  $\Phi$  is satisfiable, and let  $\phi$  be a satisfying assignment for  $\Phi$ . We construct a set  $F \subseteq V \times V$  as follows. First, for each variable  $x$  we take into  $F$  the edges  $a_{x,C}^2 a_{x,C}^3$ ,

$a_{x,C}^4 a_{x,C}^5$  for each clause  $C$  if  $\phi(x)$  is true and the edges  $a_{x,C}^1 a_{x,C}^2$ ,  $a_{x,C}^3 a_{x,C}^4$  for each clause  $C$  otherwise. Second, let  $C$  be a clause of  $\Phi$  with variables  $x$ ,  $y$ , and  $z$  and, without loss of generality, assume that the literal  $l_{x,C}$  satisfies  $C$  in the assignment  $\phi$ . For such a clause  $C$  we add to  $F$  eight elements: the edges  $q_x p_x$ ,  $q_x p_y$ ,  $q_x p_z$ , the four edges that connect  $q_y$  and  $q_z$  to the cycles  $A_y$  and  $A_z$ , and the non-edge  $q_y q_z$ .

Clearly  $|F| = \sum_{x \in \text{Vars}(\Phi)} 2s_x + 8m = k$ . We now verify that  $G \triangle F$  is a cluster graph. For each cycle  $A_x$ , the removal of the edges in  $F$  results in splitting the cycle into  $2s_x$  two-vertex clusters. For each clause  $C$  with variables  $x$ ,  $y$ ,  $z$ , satisfied by the literal  $l_{x,C}$  in the assignment  $\phi$ , the vertices  $p_x$ ,  $p_y$ ,  $p_z$ ,  $q_y$ , and  $q_z$  form a 5-vertex cluster. Moreover, since  $l_{x,C}$  is true in  $\phi$ , the edge that connects the two neighbors of  $q_x$  on the cycle  $A_x$  is not in  $F$ , thus  $q_x$  and these two neighbors form a three-vertex cluster.

**Soundness.** Let  $F$  be a minimum size feasible solution to the CLUSTER EDITING instance  $(G, k)$ . By Lemma 5, for each clause  $C$  with variables  $x$ ,  $y$ , and  $z$ , the vertices  $p_x$ ,  $p_y$ , and  $p_z$  are contained in a single cluster in  $G \triangle F$ . Denote the vertex set of this cluster by  $Z_C$ . We choose  $F$  (with minimum possible cardinality) such that the number of clusters  $Z_C$  that are contained in the vertex set  $V_C$  is maximum possible.

Informally, we are going to show that the solution  $F$  needs to look almost like the one constructed in the proof of completeness. The crucial observation is that if we want to create a six-vertex cluster  $Z_C = V_C$  then we need to put nine (instead of eight) elements in  $F$  that are incident to  $V_C$ . Let us now proceed to the formal arguments.

Fix a variable  $x$  and let  $F_x = F \cap (V(A_x) \times V(A_x))$ . We claim that  $|F_x| \geq 2s_x$  and, moreover, if  $|F| = 2s_x$  then  $F_x$  consists of every second edge of the cycle  $A_x$ . Note that  $A_x \triangle F_x$  is a cluster graph; assume that there are  $\gamma$  clusters in  $A_x \triangle F_x$  with sizes  $\alpha_j$  for  $1 \leq j \leq \gamma$ . If  $\gamma = 1$  then, as  $s_x \geq 2$ ,

$$|F_x| = |\alpha_1| = \binom{4s_x}{2} - 4s_x = 8s_x^2 - 6s_x > 2s_x.$$

Otherwise, in a cluster with  $\alpha_j$  vertices we need to add at least  $\binom{\alpha_j}{2} - (\alpha_j - 1)$  edges and remove at least two edges of  $A_x$  leaving the cluster. Using  $\sum a_j = 4s_x$ , we infer that

$$|F_x| \geq \gamma + \sum_{j=1}^{\gamma} \binom{\alpha_j}{2} - (\alpha_j - 1) = \frac{1}{2} \sum_{j=1}^{\gamma} \alpha_j^2 - 3\alpha_j + 4 = 2s_x + \frac{1}{2} \sum_{j=1}^{\gamma} (\alpha_j - 2)^2.$$

Thus,  $|F_x| \geq 2s_x$  and  $|F_x| = 2s_x$  only if for all  $1 \leq j \leq \gamma$  we have  $\alpha_j = 2$  and in each two-vertex cluster of  $A_x \triangle F_x$ ,  $F_x$  does not contain the edge in this cluster and contains two edges of  $A_x$  that leave this cluster. This situation occurs only if  $F_x$  consists of every second edge of the cycle  $A_x$ .

We now focus on a gadget for some clause  $C$  with variables  $x$ ,  $y$ , and  $z$ . Let  $F_C = F \cap (V_C \times (V_C \cup V(A_x) \cup V(A_y) \cup V(A_z)))$ . We claim that  $|F_C| \geq 8$  and there are very limited ways in which we can obtain  $|F_C| = 8$ .

Recall that the vertices  $p_x$ ,  $p_y$ , and  $p_z$  are contained in a single cluster in  $G \triangle F$  with vertex set  $Z_C$ . We now distinguish subcases, depending on how many of the vertices  $q_x$ ,  $q_y$ , and  $q_z$  are in  $Z_C$ .

If  $q_x, q_y, q_z \notin Z_C$ , then  $\{p_x, p_y, p_z\} \times \{q_x, q_y, q_z\} \subseteq F_C$  and  $|F_C| \geq 9$ .

If  $q_x \in Z_C$ , but  $q_y, q_z \notin Z_C$ , then  $\{p_x, p_y, p_z\} \times \{q_y, q_z\} \subseteq F_C$ . If there is a vertex  $v \in Z_C \setminus V_C$ , then  $F$  needs to contain three elements  $vp_x$ ,  $vp_y$ , and  $vp_z$ . In this case  $F'$  constructed from  $F$  by replacing all elements incident to  $\{q_x, p_x, p_y, p_z\}$  with all eight edges of  $G$  incident to this set

is a feasible solution to  $(G, k)$  of size smaller than  $F$ , a contradiction to the assumption of the minimality of  $F$ . Thus,  $Z_C = \{q_x, p_x, p_y, p_z\}$ , and  $F_C$  contains the eight edges of  $G$  incident to  $Z_C$ .

If  $q_x, q_y \in Z_C$  but  $q_z \notin Z_C$ , then  $q_z p_x, q_z p_y, q_z p_z, q_x q_y \in F_C$ . If there is a vertex  $v \in Z_C \setminus V_C$ , then  $F_C$  contains the three edges  $vp_x, vp_y, vp_z$  and at least one of the edges  $vq_x, vq_y$ . In this case  $F'$  constructed from  $F$  by replacing all elements incident to  $\{p_x, p_y, p_z, q_x, q_y\}$  with all seven edges of  $G$  incident to this set and a non-edge  $q_x q_y$  is a feasible solution to  $(G, k)$  of size not greater than  $F$ , with  $Z_C \subseteq V_C$ , a contradiction to the choice of  $F$ . Thus  $Z_C = \{p_x, p_y, p_z, q_x, q_y\}$  and  $F_C$  contains all seven edges incident to  $Z_C$  and the non-edge  $q_x q_y$ .

In the last case,  $V_C \subseteq Z_C$ , and  $q_x q_y, q_y q_z, q_z q_x \in F_C$ . There are six edges connecting  $V_C$  and  $V(A_x) \cup V(A_y) \cup V(A_z)$  in  $G$ , and all these edges are incident to different vertices of  $V(A_x) \cup V(A_y) \cup V(A_z)$ . Let  $uv$  be one of these six edges,  $u \in V_C$ ,  $v \notin V_C$ . If  $v \in Z_C$  then  $F$  contains five non-edges connecting  $v$  to  $V_C \setminus \{u\}$ . Otherwise, if  $v \notin Z_C$ ,  $F$  contains the edge  $uv$ . We infer that  $F_C$  contains at least six elements that have exactly one endpoint in  $V_C$  and  $|F_C| \geq 9$ .

We now note that the sets  $F_C$  for all clauses  $C$  and the sets  $F_x$  for all variables  $x$  are pairwise disjoint. Recall that  $|F_x| \geq 2s_x$  for any variable  $x$  and  $|F_C| \geq 8$  for any clause  $C$ . As  $|F| \leq 14m = 8m + \sum_x 2s_x$ , we infer that  $|F_x| = 2s_x$  for any variable  $x$ ,  $|F_C| = 8$  for any clause  $C$  and  $F$  contains no elements that are not in any set  $F_x$  or  $F_C$ .

As  $|F_x| = 2s_x$  for each variable  $x$ , the set  $F_x$  consists of every second edge of the cycle  $A_x$ . We construct an assignment  $\phi$  as follows:  $\phi(x)$  is true if for all clauses  $C$  where  $x$  appears we have  $a_{x,C}^2 a_{x,C}^3, a_{x,C}^4 a_{x,C}^5 \in F$  and  $\phi(x)$  is false if  $a_{x,C}^1 a_{x,C}^2, a_{x,C}^3 a_{x,C}^4 \in F$ . We claim that  $\phi$  satisfies  $\Phi$ . Consider a clause  $C$  with variables  $x, y$ , and  $z$ . As  $|F_C| = 8$ , by the analysis above one of two situations occur:  $|Z_C| = 4$ , say  $Z_C = \{p_x, p_y, p_z, q_x\}$ , or  $|Z_C| = 5$ , say  $Z_C = \{p_x, p_y, p_z, q_x, q_y\}$ . In both cases,  $F_C$  consists only of all edges of  $G$  that connect  $Z_C$  with  $V(G) \setminus Z_C$  and the non-edges of  $G[Z_C]$ . Thus, in both cases the two edges that connect  $q_z$  with the cycle  $A_z$  are not in  $F$ . Thus, the two neighbors of  $q_z$  on the cycle  $A_z$  are connected by an edge not in  $F$ , and  $\phi(z)$  satisfies the clause  $C$ .  $\square$

Lemma 16 directly implies the proof of Theorem 3

*Proof of Theorem 3.* A subexponential algorithm for CLUSTER EDITING, combined with the reduction shown in Lemma 16, would give a subexponential (in the number of variables and clauses) algorithm for verifying satisfiability of 3-CNF formulas. An existence of such algorithm is known to violate ETH [29].  $\square$

We note that the graph constructed in the proof of Lemma 16 is of maximum degree 5. Thus our reduction shows that sparse instances of CLUSTER EDITING where in the output the clusters are of constant size are hard.

## 6 Conclusion and open questions

We gave an algorithm that solves  $p$ -CLUSTER EDITING in time  $\mathcal{O}(2^{\mathcal{O}(\sqrt{pk})} + n + m)$ . We also have shown that the running time of our algorithm is asymptotically tight, by presenting a multivariate lower bound, and that the bound on the number of clusters is essential for subexponential tractability.

In our multivariate lower bound it is crucial that the cliques and clusters are arranged in groups of six, so that the cliques adjacent to each clause vertex are always pairwise different. However,

the drawback of this construction is that Theorem 2 settles the time complexity of  $p$ -CLUSTER EDITING problem for fixed  $p$  only for  $p \geq 6$  (Corollary 2). It does not seem unreasonable that, for example, the 2-CLUSTER EDITING problem, already NP-complete [33], may have enough structure to allow a faster algorithm, running in time subexponential in the number of vertices of the graph. Can we show such algorithm or refute its existence under ETH?

**Acknowledgements.** We thank Christian Komusiewicz for pointing us to the recent results on CLUSTER EDITING [8, 31] and his thesis [30].

## References

- [1] N. AILON, M. CHARIKAR, AND A. NEWMAN, *Aggregating inconsistent information: ranking and clustering*, in Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC 2005), ACM, 2005, pp. 684–693.
- [2] N. AILON, M. CHARIKAR, AND A. NEWMAN, *Aggregating inconsistent information: Ranking and clustering*, J. ACM, 55 (2008), pp. 23:1–23:27.
- [3] N. ALON, D. LOKSHTANOV, AND S. SAURABH, *Fast FAST*, in Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP 2009), vol. 5555 of Lecture Notes in Comput. Sci., Springer, 2009, pp. 49–58.
- [4] N. ALON, K. MAKARYCHEV, Y. MAKARYCHEV, AND A. NAOR, *Quadratic forms on graphs*, in Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC 2005), ACM, 2005, pp. 486–493.
- [5] S. ARORA, E. BERGER, E. HAZAN, G. KINDLER, AND M. SAFRA, *On non-approximability for quadratic programs*, in Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), IEEE Computer Society, 2005, pp. 206–215.
- [6] N. BANSAL, A. BLUM, AND S. CHAWLA, *Correlation clustering*, Machine Learning, 56 (2004), pp. 89–113.
- [7] A. BEN-DOR, R. SHAMIR, AND Z. YAKHINI, *Clustering gene expression patterns*, Journal of Computational Biology, 6 (1999), pp. 281–297.
- [8] S. BÖCKER, *A golden ratio parameterized algorithm for cluster editing*, in IWOCOA, C. S. Iliopoulos and W. F. Smyth, eds., vol. 7056 of Lecture Notes in Computer Science, Springer, 2011, pp. 85–95.
- [9] S. BÖCKER, S. BRIESEMEISTER, Q. B. A. BUI, AND A. TRUSS, *A fixed-parameter approach for weighted cluster editing*, in Proceedings of the 6th Asia-Pacific Bioinformatics Conference (APBC 2008), vol. 6 of Advances in Bioinformatics and Computational Biology, 2008, pp. 211–220.
- [10] S. BÖCKER, S. BRIESEMEISTER, AND G. W. KLAU, *Exact algorithms for cluster editing: Evaluation and experiments*, Algorithmica, 60 (2011), pp. 316–334.
- [11] S. BÖCKER AND P. DAMASCHKE, *Even faster parameterized cluster deletion and cluster editing*, Inf. Process. Lett., 111 (2011), pp. 717–721.
- [12] H. L. BODLAENDER, M. R. FELLOWS, P. HEGGERNES, F. MANCINI, C. PAPADOPOULOS, AND F. A. ROSAMOND, *Clustering with partial information*, Theor. Comput. Sci., 411 (2010), pp. 1202–1211.
- [13] Y. CAO AND J. CHEN, *Cluster editing: Kernelization based on edge cuts*, in Proceedings of the 5th International Symposium on Parameterized and Exact Computation (IPEC 2010), vol. 6478 of Lecture Notes in Computer Science, Springer, 2010, pp. 60–71.
- [14] M. CHARIKAR, V. GURUSWAMI, AND A. WIRTH, *Clustering with qualitative information*, in Proceedings of the 44th Symposium on Foundations of Computer Science (FOCS 2003), IEEE Computer Society, 2003, pp. 524–533.
- [15] M. CHARIKAR AND A. WIRTH, *Maximizing quadratic programs: Extending Grothendieck’s inequality*, in Proceedings of the 45th Symposium on Foundations of Computer Science (FOCS 2004), IEEE Computer Society, 2004, pp. 54–60.

- [16] J. CHEN AND J. MENG, *A  $2k$  kernel for the cluster editing problem*, Journal of Computer and System Sciences, 78 (2012), pp. 211 – 220.
- [17] P. DAMASCHKE, *Fixed-parameter enumerability of cluster editing and related problems*, Theory Comput. Syst., 46 (2010), pp. 261–283.
- [18] E. D. DEMAINE, F. V. FOMIN, M. HAJIAGHAYI, AND D. M. THILIKOS, *Subexponential parameterized algorithms on graphs of bounded genus and  $H$ -minor-free graphs*, J. Assoc. Comput. Mach., 52 (2005), pp. 866–893.
- [19] R. G. DOWNEY AND M. R. FELLOWS, *Parameterized complexity*, Springer-Verlag, New York, 1999.
- [20] M. R. FELLOWS, J. GUO, C. KOMUSIEWICZ, R. NIEDERMEIER, AND J. UHLMANN, *Graph-based data clustering with overlaps*, Discrete Optimization, 8 (2011), pp. 2–17.
- [21] J. FLUM AND M. GROHE, *Parameterized Complexity Theory*, Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, Berlin, 2006.
- [22] F. V. FOMIN AND D. KRATSCHE, *Exact Exponential Algorithms*, An EATCS Series: Texts in Theoretical Computer Science, Springer, 2010.
- [23] F. V. FOMIN AND Y. VILANGER, *Subexponential parameterized algorithm for minimum fill-in*, in Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2012), SIAM, 2012, pp. 1737–1746.
- [24] I. GIOTIS AND V. GURUSWAMI, *Correlation clustering with a fixed number of clusters*, in Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2006), ACM Press, 2006, pp. 1167–1176.
- [25] J. GRAMM, J. GUO, F. HÜFFNER, AND R. NIEDERMEIER, *Graph-modeled data clustering: Exact algorithms for clique generation*, Theory Comput. Syst., 38 (2005), pp. 373–392.
- [26] J. GUO, *A more effective linear kernelization for cluster editing*, Theor. Comput. Sci., 410 (2009), pp. 718–726.
- [27] J. GUO, I. A. KANJ, C. KOMUSIEWICZ, AND J. UHLMANN, *Editing graphs into disjoint unions of dense clusters*, Algorithmica, 61 (2011), pp. 949–970.
- [28] J. GUO, C. KOMUSIEWICZ, R. NIEDERMEIER, AND J. UHLMANN, *A more relaxed model for graph-based data clustering:  $s$ -plex cluster editing*, SIAM J. Discrete Math., 24 (2010), pp. 1662–1683.
- [29] R. IMPAGLIAZZO, R. Paturi, AND F. ZANE, *Which problems have strongly exponential complexity?*, J. Comput. Syst. Sci., 63 (2001), pp. 512–530.
- [30] C. KOMUSIEWICZ, *Parameterized Algorithmics for Network Analysis: Clustering & Querying*, PhD thesis, Technische Universität Berlin, 2011. Available at <http://fpt.akt.tu-berlin.de/publications/diss-komusiewicz.pdf>.
- [31] C. KOMUSIEWICZ AND J. UHLMANN, *Alternative parameterizations for cluster editing*, in SOFSEM, I. Cerná, T. Gyimóthy, J. Hromkovic, K. G. Jeffery, R. Královic, M. Vukolic, and S. Wolf, eds., vol. 6543 of Lecture Notes in Computer Science, Springer, 2011, pp. 344–355.
- [32] F. PROTTI, M. D. DA SILVA, AND J. L. SZWARCFITER, *Applying modular decomposition to parameterized cluster editing problems*, Theory Comput. Syst., 44 (2009), pp. 91–104.
- [33] R. SHAMIR, R. SHARAN, AND D. TSUR, *Cluster graph modification problems*, Discrete Applied Mathematics, 144 (2004), pp. 173–182.